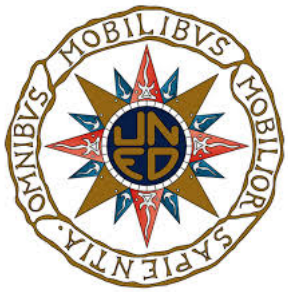


Máster Universitario en Investigación en Ingeniería de Software y Sistemas Informáticos

Modelos simbólico-conexionistas para la segmentación y descripción de Marcas de Cantero

ITINERARIO DE INGENIERÍA DE SOFTWARE | COD. ASIGNATURA: 105128

ALUMNO

CARLOS JIMÉNEZ DE PARGA

DIRECTOR

D. CARLOS CERRADA SOMOLINOS

Departamento de Ingeniería del Software y Sistemas Informáticos

CURSO: 2013-2014 || DEFENSA JUNIO

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA (UNED)

MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS

Modelos simbólico-conexionistas para la segmentación y descripción de Marcas de Cantero

ITINERARIO DE INGENIERÍA DE SOFTWARE | COD. ASIGNATURA: 105128

TRABAJO PERTENECIENTE AL ÁREA DE SISTEMAS DE PERCEPCIÓN VISUAL

ALUMNO

CARLOS JIMÉNEZ DE PARGA

DIRECTOR

D. CARLOS CERRADA SOMOLINOS

Departamento de Ingeniería del Software y Sistemas Informáticos

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA (UNED)

CALIFICACIÓN: _____

(A cumplimentar por secretaría)

Impreso TFdM05_Autor. Autorización de publicación

y difusión del TFdM para fines académicos

Autorización

Autorizo/amos a la Universidad Nacional de Educación a Distancia a difundir y utilizar, con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firma del/los Autor/es

Juan del Rosal, 16
28040, Madrid

Tel: 91 398 89 10
Fax: 91 398 89 09

www.issi.uned.es

RESUMEN:

La presente tesis de fin de máster pretende ilustrar las técnicas basadas en formalismos matemáticos y bio-inspirados, utilizando algoritmos de procesamiento de imágenes con la finalidad de aplicarlas al reconocimiento de marcas de cantero. Bajo unos principios y una metodología investigadora tecno-científica se han diseñado y programado una serie de algoritmos simbólicos y conexionistas a un conjunto de signos escogidos de diferentes ubicaciones de nuestra geografía. La dificultad de discriminación de la propia marca de cantería con respecto al ruido de fondo de los sillares ha sido la principal piedra de toque de la evaluación de las diferentes técnicas y de las cuales se reseñan en el presente estudio. El uso de herramientas de edición gráficas junto al software matricial Matlab han permitido la aplicación práctica de los modelos que se presentan a continuación.

LISTA DE PALABRAS CLAVE:

Marcas de cantero, ImageJ, Matlab, Investigación, Viollet-Le-Duc, Lampérez y Romea, Signos lapidarios, Sillar, Segmentación, Descripción, Colorimetría, Catedral, Ruido, Bordes, Sobel, Gimp, Mapa de relieve, Filtros, Búsqueda en profundidad, JPEG, PNG, Escala de grises, Blanco y negro, Binarización, Escalado, Rotación, Ecualización, Brillo, Contraste, Simbólico, Conexionista, Perceptrón, Aprendizaje supervisado, Aprendizaje no supervisado, Redes de Hopfield, Redes neuronales, Feed-forward, Back-propagation, Sigmoide, Función de activación, Tangente Hiperbólica, Función de entrenamiento, Neurona, Capa, Multiclase, Tasa de aprendizaje, Derivable, Epochs, Show, Goal, Algoritmo, Error cuadrático medio, Lineal, No lineal, Rendimiento, Tolerancia, Pentagrama, Cruz, Correlación Estadística, Estadística Descriptiva, Desviación típica, Covarianza, Coeficiente de correlación, Media, Código de cadena, Eficacia, Juego de pruebas, Porcentaje, Pésimo, Óptimo,Estudio.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

MOTIVACIÓN

Y ateniéndonos a la siguiente reseña:

El presente volumen reúne todos los artículos de René Guénon sobre los símbolos tradicionales que aparecieron en publicaciones periódicas entre 1925 y 1950. En ellos se explica que el símbolo tiene un origen no humano y se basa en la correspondencia entre dos órdenes de realidades: no expresa ni explica, sino que sólo sirve de soporte para elevarse, mediante la meditación, al conocimiento de las verdades metafísicas. Teniendo en cuenta todo esto, el autor llega a la conclusión de que la decadencia de los símbolos que caracterizan a nuestra época es el resultado de la pérdida de mentalidad simbólica, de su degradación, de la incomprensión de su sentido y de su estudio meramente exterior. Y de aquí que, a la filosofía profana, que emplea un lenguaje analítico y racional (el naturalismo, el materialismo...), Guenón acabe oponiendo la metafísica, ciencia sagrada, que usa un lenguaje sintético y espiritual: precisamente el simbolismo.

"Símbolos Fundamentales de la Ciencia Sagrada. René Guénon — 1962 Éditions Gallimard, París."

---

1. INTRODUCCIÓN

Comenzamos esta andadura por el análisis gliptográfico exponiendo los principales objetivos por los que ha merecido la pena abordar el presente trabajo de fin de máster. Si bien en la sociedad actual hay un cierto y justificado descontento al plano espiritual y metafísico, es cierto que el fomento de la diversidad ideológica moderna ha auspiciado iniciativas dirigidas al fomento de la diversidad cultural y la protección del bien cultural común. Esto viene justificado por el reciente fenómeno de la globalización y la crisis de la nueva sociedad. El hecho de promover iniciativas dirigidas hacia la cultura en todas sus facetas se ha convertido en un remedio, cura y salvaguarda de valores e ideas, ante una sociedad decadente originada por el fracaso del neocapitalismo, la incultura, la hipertecnología y la degradación moral de las masas.

Por este motivo, la investigación dirigida al plano cultural puede ayudar, en cierta medida, ha crear una base sólida de valores para las futuras generaciones. Por tanto, el desarrollo en un plano científico y epistemológico de un estudio de las marcas de cantero con técnicas digitales de procesamiento de la imagen, pueden ayudar a los arqueólogos y estudiosos del arte para generar iniciativas en este orden de ideas.

El estudio de las marcas de cantero no solo sirve como excusa para aplicar fríamente unas técnicas científicas de procesamiento de imágenes, sino que intenta ayudar a la sociedad a comprender mejor su pasado, su origen humano, su historia y la trascendencia simbólica y metafísica que estos anónimos operarios han dejado grabadas para siempre en las piedras de nuestras catedrales.

A lo largo de estas páginas encontrará la descripción formal de los diferentes algoritmos utilizados para segmentar y describir marcas de cantero utilizando los conocimientos aprendidos en el máster de Ingeniería de Software y Sistemas Informáticos de la UNED. El estudio/investigación e implementación de algoritmos e interfaces de usuario ha llevado una cantidad de tiempo que es equiparable al número de créditos ECTS establecidos para el trabajo de fin de máster, por lo que se ha intentado, en la mayor medida de lo posible, ajustarse a un trabajo de excelencia investigadora de cara a la realización de una futura tesis de doctorado.

El estudio de la asignatura de *Sistemas de Percepción Visual* ha facilitado la investigación y aplicación de algoritmos. La utilización conjunta de Matlab con herramientas como Gimp e ImageJ han servido para el estudio detallado de cada una de las marcas de cantero. La incorporación de un lenguaje de programación y la inclusión de Redes Neuronales en las librerías de Simulink me han ayudado a investigar el comportamiento de estos sistemas conexionistas.

2. OBJETIVOS

En esta sección se van a estudiar los diferentes objetivos con los que se pretende abordar el trabajo y alcanzar un criterio de calidad en el resultado obtenido.

2.1. OBJETIVOS TEÓRICOS

Obviamente, el presente trabajo de fin de máster, al estar encuadrado en la asignatura de *Sistemas de Percepción Visual* de 9 créditos anual del perfil de Sistemas Informáticos, se encuentra estrechamente relacionada con los contenidos explicados en el libro oficial de la asignatura: "Visión por computador"de Gonzalo Pajares y Jesús M. de la Cruz García. [1].

Por este motivo, y antes comenzar la explicación de las diferentes técnicas, se hace necesario estudiar y examinarse de los dos bloques temáticos encuadrados dentro del temario de la asignatura:

Unidad temática	
UT-01. Visión Artificial. UT-02. Geometría y parámetros de las cámaras. UT-03. Fundamentos del color UT-04. Transformaciones matemáticas. UT-05. Prepocesamiento de imágenes. UT-06A. Extracción de características-I	1ª TAREA EVALUABLE
UT-06B. Extracción de características-II UT-07. Secuencias de imágenes UT-08. Reconocimiento de patrones. UT-09. Descripción y reconocimiento de objetos 3D. UT-10. Obtención de la forma.	2ª TAREA EVALUABLE

Cuadro 2.1: Tabla de contenidos a estudiar

Una vez que el alumno ha estudiado y realizado las dos tareas evaluables y obligatorias se debe enfrentar a un examen telemático de baja-media complejidad donde debe demostrar sus conocimientos teóricos y destrezas en la resolución de problemas de Visión Artificial. Esto le permitirá adquirir casi con toda seguridad unos conocimientos teóricos que podrá aplicar en el campo del análisis digital de imágenes.

Para complementar los conocimientos teóricos previamente exigidos, el alumno se debe enfrentar posteriormente a una herramienta profesional para el tratamiento de imágenes digitales. El desarrollo de una memoria explicando la aplicación práctica de varios ejemplos de los siguientes contenidos será suficiente para comprobar su nivel de conocimientos prácticos:

Contenidos	Bloque temático
Representación y tratamientos básicos	Bloque 1
Transformaciones	Bloque 2
Transformaciones geométricas	Bloque 3
Color	Bloque 3.2
Correcciones radiométricas	Bloque 3.3
Extracción de bordes	Bloque 4.1
Extracción de regiones	Bloque 4.2
Puntos de interés	Bloque 4.3
Descripción de bordes	Bloque 5.1
Descripción de regiones	Bloque 5.2
Operaciones morfológicas	Bloque 6

Cuadro 2.2: Tabla de bloques prácticos exigidos por la UCM

2.2. OBJETIVOS TÉCNICOS

Los objetivos técnicos del trabajo son básicamente realizar una herramienta visual en Matlab donde el usuario pueda leer una imagen de un fichero gráfico y representarla en pantalla. Posteriormente, y con la ayuda de los comandos del menú de la interfaz de usuario, el gliptógrafo podrá aplicar las técnicas de síntesis de imagen para filtrar, segmentar y comparar con una base de datos externa.

La idea es utilizar las siguientes herramientas:

Lenguaje	Matlab 8.01
Sistema operativo	Linux Ubuntu 12 64-bit
Sistema	PC - Intel Core i7 - NVidia GForce9500-GT
Pantalla	LCD - LG - 24 pulgadas panorámica
Herramientas gráficas	GIMP 2.8
Herramientas avanzadas	ImageJ
Lenguaje de soporte	Java 1.7
Captura de imágenes	Scanner Brother - iPhone 4s (cámara)
Herramientas matemáticas	Matlab 8.01 y Calculadora Casio fx-115MS

Cuadro 2.3: Tabla de requisitos técnicos

Con todas estas herramientas se pretende realizar el análisis de los modelos y la programación del los algoritmos.

2.3. OBJETIVOS PERSONALES

Mi objetivo personal es conseguir una herramienta útil y eficiente para el historiador o el gliptólogo interesado en estudiar el fenómeno de las marcas de cantero. Mi idea es liberar la aplicación como Open Source para su posterior utilización en el mundo profesional. Con estos objetivos en mente, aspiro a conocer mejor las técnicas de tratamiento digital de imágenes que he estudiado. También pretendo ampliar mis conocimientos en un tema que me apasiona como es el del arte medieval, aplicándolos a los estudios que con tanto esfuerzo he dedicado durante toda mi vida.

3. METODOLOGÍA

3.1. METODOLOGÍA DE INVESTIGACIÓN

La investigación que se va a aplicar al proyecto es del tipo de *Investigación aplicada* puesto que es la utilización de los conocimientos en la práctica, para aplicarlos, en la mayoría de los casos, en provecho de la sociedad. Para ello se ha aplicado el *método científico* del tipo *empírico-analítico* el cual se basa en la empírica y en la medición experimental autocorrectiva-progresiva, sujeto a los principios específicos de las pruebas de razonamiento. Según el diccionario de Oxford, el método científicos es: "un método o procedimiento que ha caracterizado a la ciencia natural desde el siglo XVII, que consiste en la observación sistemática, medición y experimentación, y la formulación, análisis y modificación de las hipótesis".

El método científico se basa en dos principios fundamentales:

1. **La reproducibilidad**: Es decir, la capacidad de repetir un determinado experimento, en cualquier lugar y por cualquier persona. Este principio se basa, esencialmente, en la comunicación y publicidad de los resultados obtenidos (por ejemplo en forma de artículo científico).
2. **La refutabilidad**: Esta principio explica que toda proposición científica tiene que ser susceptible de ser falsada o refutada. Esto implica que se podrían diseñar experimentos, que en el caso de dar resultados distintos a los predichos, negarían la hipótesis puesta a prueba. La falsabilidad no es otra cosa que el *modus tollendo tollens*[1] del método hipotético deductivo experimental.

Por tanto el método de investigación seguirá las siguientes etapas:

1. Se determinará el tipo de investigación a realizar.
2. Se determinarán las fuentes de datos a recolectar.
3. Se determinará el diseño de la investigación.
4. Se determinará la muestra de imágenes de marcas de cantero.

[1] $(p \longrightarrow q), \neg q \vdash \neg p$

5. Se procederá al diseño de los algoritmos de segmentación y descripción sobre muestras.
6. Se interpretarán los resultados hasta la satisfacción de los algoritmos.

De acuerdo con Malhotra(1997) el diseño de la investigación es una estructura o un plano que sirve para dirigir un proyecto de investigación. Nos detalla los pasos necesarios para obtener información indispensable en la solución al problema que nos ocupa.

El diseño de la investigación incluye los siguientes pasos:

1. Se consultará la bibliografía necesaria.
2. Se buscará las fuentes de los autores de algoritmos semejantes en Internet.
3. Se realizará un análisis de los datos obtenidos con ImageJ y Gimp.
4. Se consultará a expertos.
5. Se esbozará un prototipo de algoritmo en Matlab.
6. Se comprobará su eficiencia y eficacia.
7. Se desarrollará una aplicación integradora de todas las técnicas en Matlab.

3.2. METODOLOGÍA GLIPTOGRÁFICA

Según algunos investigadores y [11] se puede establecer una base de estudio de marcas a partir de lo siguientes criterios:

1. La época de las diferentes partes de un monumento.
2. Localización interior o exterior en ábsides, muros, pilares, arcadas, dovelas, etc.
3. Estadística de frecuencia y variación de las medidas.
4. Aparición de dos o más signos sobre el mismo sillar.
5. Trazos rectos o curvos y si éstos son finos, normales o profundos.
6. Clase de signos: letras, objetos, siluetas de animales, especialmente aves, figuras definidas, dibujos geométricos y formas sin identificar.
7. Significado probable de su origen:

    -Letras iniciales, anagramas o nombres completos.

    -Signos astrológicos.

    -Signos alquímicos.

    -Signos numéricos.

    -Signos místico-cristianos.

-Signos representativos de logias o asociaciones.

-Signos sobre un estado social o profesión.

-Signos de instrumentos de cantería.

-Signos expresivos de una nacionalidad.

-Signos relacionados con donantes.

-Signos a los que no cabe adscribir significado u orden concreto.

Esta clasificación permite hacer una lectura inicial y elemental de cualquier marca de cantería que se pueda encontrar y cuyo análisis puede confirmarnos algunos detalles:

a) El origen religioso, civil o militar de los restos de un monumento. Las marcas en edificios de carácter militar son muy numerosas y suelen tener trazos poco elegantes y con formas caprichosas y rudas. En los edificios civiles son menos frecuentes y casi inexistentes. Sin embargo, los edificios religiosos son los que más variedad de formas ostentan, abundando las cruces y las formas elegantemente geométricas.

b) Las logias a los que obreros pertenecían. Este aspecto ha sido clave para el desarrollo principal del trabajo, puesto que esta hipótesis se basa en que ciertas marcas pertenecían a un cantero itinerante o a una logia concreta. Por lo tanto, dirigir la investigación hacia la identificación descriptiva de la marca con respecto a la realizada por la misma logia u operario en otra obra ha sido el objetivo principal.

c) La forma de estar tallado el signo, su profundidad y técnica de acabado. Existen marcas muy deterioradas (algunas por el paso del tiempo), aunque en muchas ocasiones es debido a la habilidad del tallista para hacer el dibujo. Cuando encontramos una marca perfectamente definida y un acabado nítido, como si hubieran sido realizadas recientemente, podemos estar casi seguros que nos encontramos ante un cantero de alta escuela de gran categoría en su profesión.

d) El estatus social, pasado o presente, del tallista. Se admite que ciertos signos que representan herramientas (hacha, mazo, nivel, compás, etc.) son indicativos del oficio presente del obrero. En otros casos, representaría su condición social pasada, como es el caso del signo que tiene forma de ballesta, muy numeroso en distintos monumentos, indicativo de la profesión anterior de soldado, o la letra S mayúscula, dividida verticalmente por una línea, que definiría un estado anterior de esclavitud.

e) El nombre del obrero en forma de monograma, anagrama o, más raro, el nombre y apellido completo.

f) La repetición de un signo en un mismo edificio.

g) El número de marcas diferentes que se pueden catalogar en un monumento prueba el número de obreros que trabajaron.

h) Ciertas creencias personales del tallista que se pueden rastrear en determinados signos de contenido místico, alquímico u ocultista: las estrellas de cinco puntas o seis puntas, por ejemplo.

4. MARCAS DE CANTERO

4.1. HISTORIA

Las marcas de cantero se remontan a tiempos muy pretéritos cuando los artesanos de las civilizaciones caldea, egipcia, persa, griega y romana marcaban las construcciones utilizando una simbología aún desconocida hasta la fecha.

En general, las marcas de cantería que aparecen en sillares, columnas , bóvedas pertenecen a gremios de constructores de todas las épocas, por ejemplo, los artesanos de los *collegia fabrorum* romanos o los canteros medievales.

Históricamente las marcas de cantero nunca llamaron la atención de visitantes ni de estudiosos de otras épocas, hasta que a mediados del siglo XIX, gracias entre otros autores a *Viollet-Le-Duc*, un arquitecto del romanticismo francés que dedicó su vida al estudio y reconstrucción de la catedrales góticas francesas, centrara su atención y análisis a estos curiosos símbolos grabados en la piedra de estas construcciones.

Lampérez y Romea en su extensa monografía sobre arquitectura cristiana, destaca que la información acumulada sobre los signos lapidarios, marcas de cantero, masónicos y francmasónicos es pobrísima[2]. Además la oscuridad sobre este tema es debido a la carencia de tratados monacales y de la Edad Media sobre este tipo de signos que es prácticamente escasa. Con todo ello y a pesar de los intentos de los últimos años en recavar información sobre este fenómeno, es evidente la dificultad para investigar ampliamente este campo.

La hipótesis más aceptada, expuesta por *M.Dridon* y *Viollet-le-Duc*, explica las marcas de cantero como signos lapidarios pertenecientes a la categoría de signaturas personales de los canteros, aparejadores y maestros de obra que, en muchos casos, servían para señalar la faena realizada y determinar el salario correspondiente.

Las notaciones más características utilizadas suelen ser monogramas o signos geométricos más o menos complicados, basados en la simbología de monumentos griegos, romanos y bizantinos, aunque los del período medieval ofrecen mayor complejidad, puesto que aquellos se limitan a formas muy simples. Es importante recalcar que no todas las marcas encontradas tienen que ser de canteros, pueden

[2] Firmado en la piedra. Véase [11]

Figura 4.1: Marca de cantero en la Concatedral de Soria

ser, sin embargo, de trazos dejados por los compañeros de las logias de canteros en sus viajes, marcas de obreros, graffitis de peregrinos, vítores, etc.

4.2. SIGNIFICADO E INTERPRETACIONES

Entre las interpretaciones e hipótesis del origen de las marcas cantero, *Lampérez y Romea* ha resumido cinco de las más importantes:

a) **Los signos lapidarios son el alfabeto de un lenguaje mágico y esotérico, con origen en la magia caldea y empleado como conjuro contra las potencias enemigas y de la Naturaleza.** Esta hipótesis se fundamenta en el empleo desde muy antiguo, como el el caso de la esvástica o cruz de los arios, el sello de Salomón o estrella de seis puntas, el microcosmos o figura pitagórica y el pentalfa o estrella de cinco puntas. Se fundamenta que estos signos se han utilizado a través de los tiempos ubicándolos en lugares sagrados como edificios, basas de las columnas y primeras hiladas de los zócalos, rosetones, etc. Las críticas a esta teoría señalan que, aunque estos signos existen, son escasos y podrían deberse a logias o corporaciones de maestros y obreros que estaban muy relacionadas con geómetras y alquimistas, con la siguiente mezcla se signos. No hay que descartar que algunos de ellos, a pesar de ser usados por ocultistas caldeos y por otras culturas, hallan perdido su carácter iniciático, de la misma forma que muchos diseños basados en signos lapidarios utilizados hoy día por sociedades secretas, no tienen nada que ver con el significado técnico que mantenían en la Edad Media [11]. Otro problema para admitir esta hipótesis es la permisividad de la Iglesia para dejar marcar los muros con signos de origen pagano.

b) **Los signos lapidarios son marcas hechas por los canteros para el ajuste y asiento respectivo de los sillares**. Esta es una de las teorías más aceptadas y más sólidas. Como se comentó anteriormente, esta hipótesis se fundamenta en los conceptos constructivos de iglesias y catedrales en la Edad Media, ya que el ensamblaje de los sillares y piezas de piedra se marcaban con signos de ubicación a modo de manual de instrucciones.

c) **Las marcas de cantero son las firmas y signaturas de cada cual, para facilitar la posterior liquidación y cobro del trabajo**. Esta teoría fue apadrinada por *Didron* y *Viollet-le-Duc*, es una de las que han gozado de más popularidad. Una crítica a esta hipótesis es el caso de una gran cantidad de marcas que no responden a esta posibilidad, ya que la interpretación de su simbología no aparenta representar una indicación de este tipo.

d) **Las marcas de cantero son marcas personales de cada cantero referentes a su nombre (en forma inicial o monograma), a sus creencias o devociones (un objeto simbólico o alegórico), a su estado social o profesión pasada o presente (un signo de esclavitud o un útil) o la época en que se labró la obra (un signo astrológico, etc.)** La aceptación de esta teoría no excluye a la anterior, y se admite que ambas pueden ser simultáneas. La señal de una personalidad concreta y su origen gremial a través del tiempo es la hipótesis más plausible.

e) **Los signos lapidarios podrían ser, en algunas ocasiones, conjunta e independientemente de algunas de las anteriores significaciones, una signatura del donante de un sillar, de una columna, de una bóveda,etc.** Esta hipótesis aclara la

posibilidad de que ciertas corporaciones o donantes (reyes, nobles, etc.) sufragaran parte de la construcción de la obra.

4.3. UBICACIONES PARA EL ESTUDIO

Las marcas cantero se encuentran ampliamente extendidas por toda Europa, Medio Oriente y norte de África. En España, en concreto, se encuentran ubicadas en prácticamente toda la península, pero fundamentalmente en el norte y la parte de los Caminos de Santiago.

Para el presente estudio se ha a recurrido a fotografías extraídas de las siguiente ubicaciones mostradas en el mapa:

Figura 4.2: Ubicaciones para el estudio

Estas ubicaciones han sido visitadas por el alumno en los últimos años, por lo que el origen de las fotografías se encuentra en la siguientes localizaciones:

1. **León**

   Camino de Santiago

   Catedral de León

   San Isidoro de León

2. **Soria**

   Concatedral de San Pedro de Soria

   San Juan de Duero

3. **Navarra**

   Camino de Santiago

   Catedral de Pamplona

4. **Murcia**

   Catedral de Murcia

5. **Tarragona**

   Catedral de Tarragona

El conjunto de fotografías es bastante heterogéneo y su tipología se encuentra dentro de la clasificación del apartado anterior.

Para el estudio de los diferentes algoritmos de segmentación con las herramientas anteriormente expuestas se han utilizado varias fotografías de marcas de cantero; no obstante solo un conjunto reducido ha sido seleccionado para la representación de los ejemplos en el software de demostración.

5. PROBLEMÁTICA

En esta sección se realizará un análisis preliminar a la problemática subyacente en el procesamiento de marcas de cantero. En general abarca la detección de bordes de demarcación de figura geométrica y la eliminación de ruido de fondo. Respecto a este último la problemática asociada a las técnicas de detección de poros, surcos y daño en la piedra es un aspecto clave para la investigación.

5.1. IMAGEN ORIGINAL. EJEMPLOS.

Nos encontramos en la fase de adquisición de imágenes después del fotografiado de las marcas en las ubicaciones expuestas en el apartado anterior:

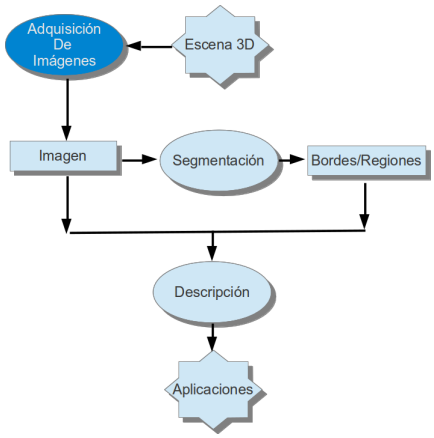


Figura 5.1: Fase de adquisición

Las marcas de cantero se realizan sobre piedra de tipo granítico, calizo, etc. con martillo y cincel. El objetivo operario es realizar un surco en la piedra a modo de figuras geométricas básicas o más complejas. Obviamente si se observa la marca con una lupa se puede observar que dichas líneas o curvas de base geométrica tienen una irregularidad considerable. Por ejemplo:

Si observamos esta imagen ampliándola:

Figura 5.2: Marca de cantero de la Catedral de Murcia

Figura 5.3: La figura es irregular en su estructura macroscópica

En la figura 5.4 podemos observar cómo las líneas curvas son irregulares en la ampliación.

Esta irregularidad del contorno se debe en mayor parte a la porosidad de la piedra, causada por el paso del tiempo sobre el material de la misma. Como veremos en el apartado de investigación de la extracción de códigos de cadena con Matlab, esta irregularidad que conforman los poros nos ayudará a crear un algoritmo estadístico para extraer la forma o código de cadena al cambiar los matices de los colores del surco.

Figura 5.4: Marca de cantero del claustro de la Concatedral de Soria ampliada

5.2. COLORIMETRÍA DE LAS MARCAS DE CANTERO

Nos encontramos en la fase de segmentación:

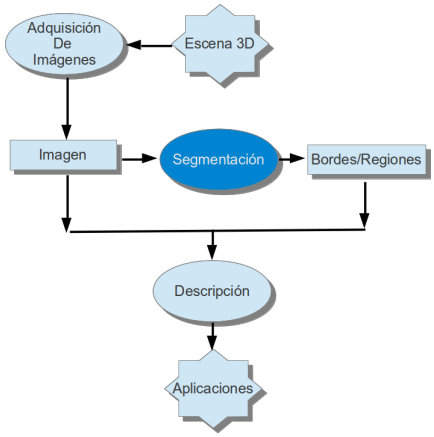


Figura 5.5: Fase de segmentación

Si analizamos los colores del espectro electromagnético contenidas en cada una los sillares en donde se han realizado las marcas observamos el siguiente conjunto:

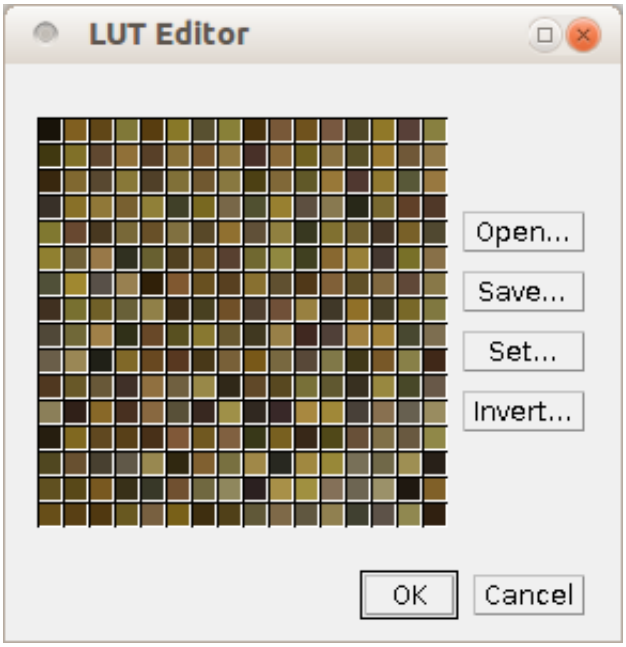


Figura 5.6: Colores de la Marca de cantero de la Catedral de Murcia5.17

Se puede apreciar que los matices verdosos, marrones y amarillentos recaen en la piedra, mientras que los colores más oscuros (verde y marrón oscuro) pertenecen a los poros y a los surcos.

Veamos ahora un análisis de las componentes R, G, B:

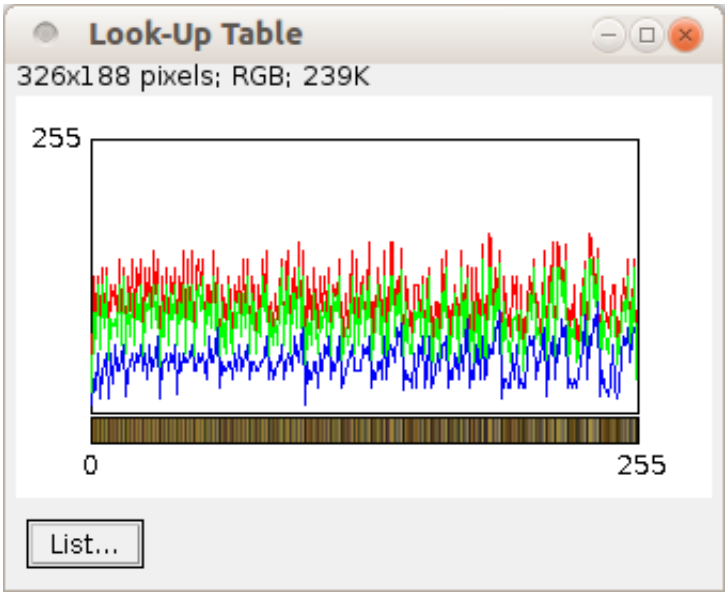


Figura 5.7: Componentes RGB de la marca de Murcia5.17

Donde se puede apreciar un predominio de las componentes roja y verde, fundamentalmente el rojo.

Si analizamos la marca de la Concatedral de Soria:

apreciamos de nuevo una gama de colores marrones, verdosos, rojizos y grises.

En el análisis de las componentes RGB apreciamos la siguiente información:

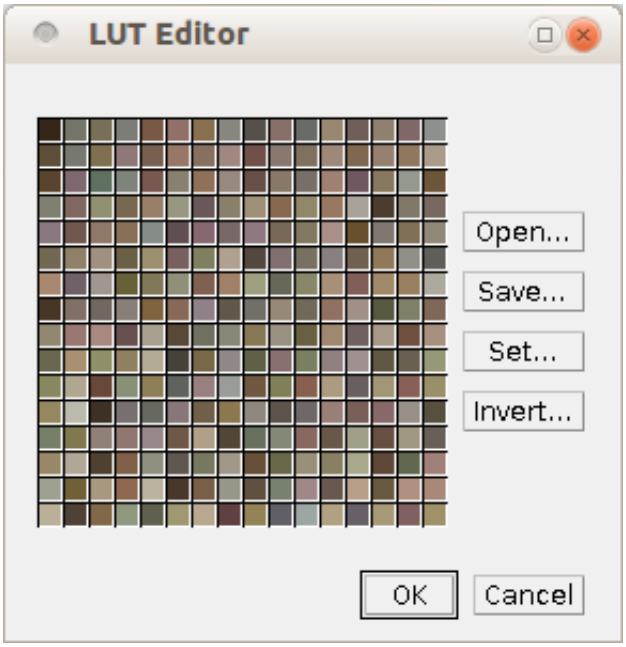


Figura 5.8: Colorimetría de la marca de la Concatedral de Soria5.4

Figura 5.9: Componentes RGB de la Concatedral de Soria5.4

Se aprecia una gama de componentes más equilibrada lo que indica que el día

que se hizo la foto el cielo estaba nublado, por lo que la tendencia es a unos colores centrados en el gris.

La conclusión que se obtiene de este análisis indica que la gama de colores de una foto de una marca de cantero suele ser muy homogénea, dejando muy poco espacio para la discrepancia de colores. Como se ha visto en las dos colorimetrías de ImageJ, la tendencia es a no distanciarse mucho del color marrón y gris, quedando la imagen centrada sobre estos matices. Obviamente, esta información es útil a la hora de segmentar la imagen para eliminar ruido y extraer las regiones y los bordes (principalmente en el algoritmo investigado para los códigos de cadena), ya que las intensidades oscuras de los matices marrones y grises nos pueden proporcionar una pista en cuanto a la presencia de una Marca de cantero. No obstante, se presenta un nuevo desafío a la hora de discriminar entre poros de la piedra y surcos tallados por el masón; lo que nos indica que quizá la técnica colorimétrica no sea suficiente para resolver dicho problema.

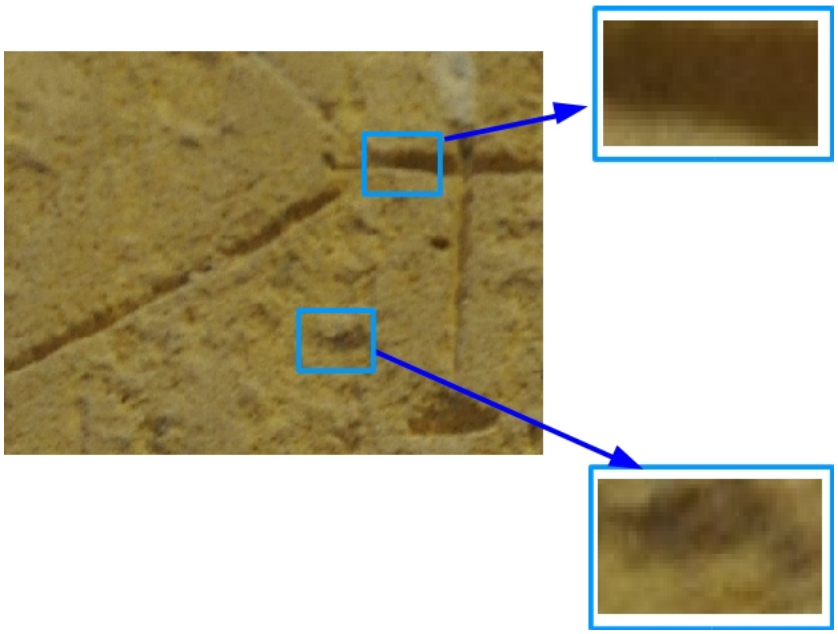


Figura 5.10: Ampliaciones de marca y poro en la Catedral de Murcia5.17

donde los espectro colorimétricos son para:

A) **B)**

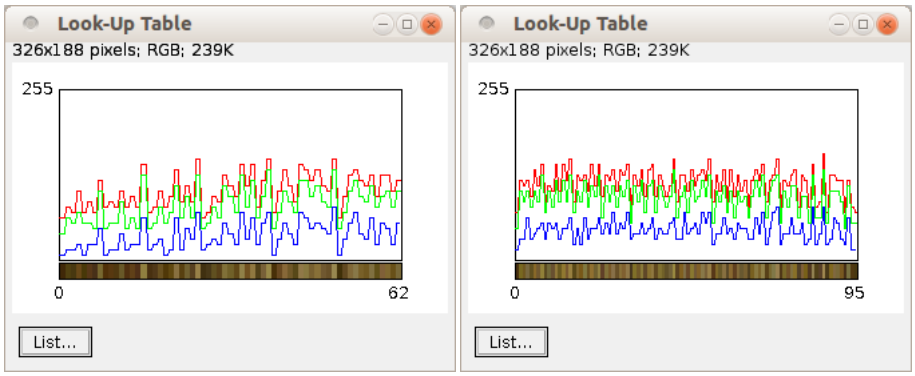


Figura 5.11: Espectro colirmétrico de la marca(izquierda) y poro (derecha)5.17

5.3. RUIDO Y BORDES

La mayor problemática a la hora de procesar las imágenes de las marcas de cantero recae en la técnica a aplicar para extraer los bordes para un posterior procesamiento de los códigos de cadena. La idea sería la de utilizar directamente alguna técnica de extracción de bordes por primera o segunda derivada, operadores de Sobel, Prewit, etc.

No obstante si realizamos dicha operación con GIMP, obtenemos un resultado no satisfactorio:

Figura 5.12: Marca de cantero de la Catedral de Murcia

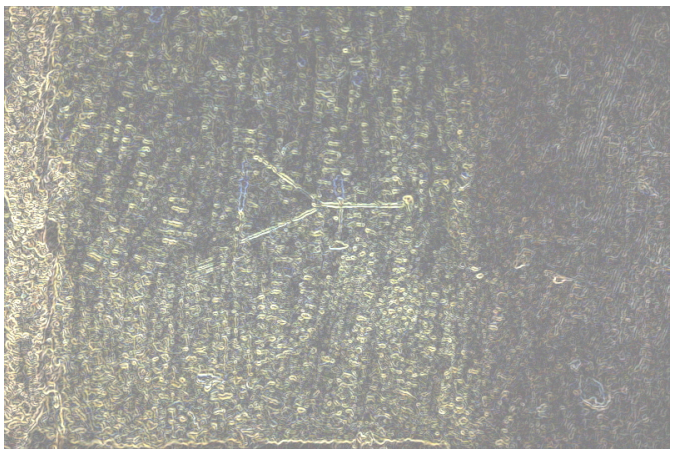


Figura 5.13: Marca de cantero de la Catedral de Murcia procesada por Sobel

Como se puede apreciar, el propio surco de la marca se confunde con el ruido de la piedra. Esto repercute considerablemente a la hora de diseñar un algoritmo efectivo que discrimine entre el surco de la marca y los poros de la piedra [3]. Una posible técnica para discriminar entre los poros y las marcas podría consistir en dividir la imagen en cuadrículas de igual tamaño y aplicar una algoritmo que realice la media de la celda, si ésta tiene unos valores promedio cercanos al marrón oscuro se consideraría como una celda a considerar como marca, en caso contrario se descartaría.

Retomando de nuevo las operaciones de extracción de bordes, vemos cómo los operadores de Sobel implementados en GIMP producen un resultado confuso para aplicar el algoritmo de cadenas. Adicionalmente se observa que al aplicar un operador de extracción de bordes se pierde información de las microformas de la propia marca (véase la parte superior de la cruz formada en la marca de la figura 5.12). Esto nos lleva a un nuevo problema en el caso de recurrir a los operadores de extracción de bordes pues el resultado de la extracción de un borde se confunde aún más con el poro.

Una posible solución proporcionada por GIMP es el *filtro de relieve*. Éste nos da la posibilidad de aumentar las diferencias de relieve en la propia piedra, creando así un surco más profundo que puede ser mejor detectado por el algoritmo de extracción de códigos de cadena.

Veamos ahora un ejemplo de aplicación de dicho filtro:

[3] Como se vio en el análisis colorífico, la diferencia de colores entre una marca y un poro es casi inapreciable.

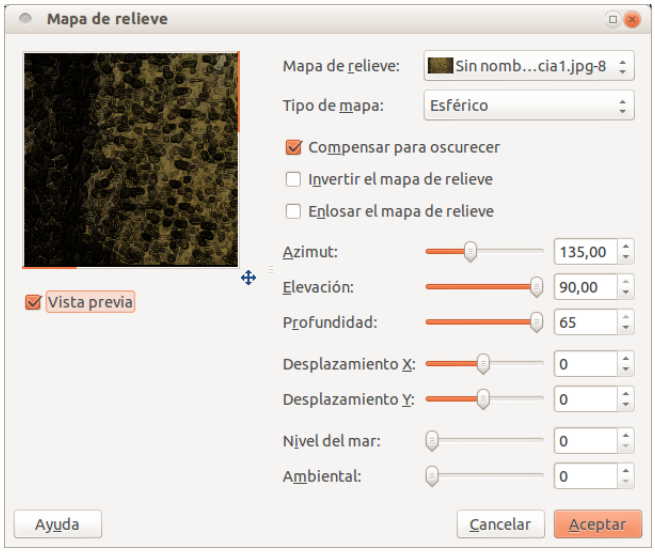


Figura 5.14: Opciones del filtro de relieve

Dicho filtro se accede en GIMP mediante **Filtros->Mapa->Mapa de relieve** y posicionando la Elevación y la Profundidad al máximo, e indicando un tipo de mapa Esférico. El resultado será el siguiente:

Figura 5.15: La marca de Murcia aumentada en relieve

Como se puede apreciar, al aplicar este filtro se pronuncian los valles de los surcos de la piedra para tallar la marca. Esto facilitará las cosas para el algoritmo de extracción de los códigos de cadena que tendrá ahora más información. Pero previo al paso de la extracción de los códigos de cadena será necesario amplificar aún más la señal aumentada del relieve de la marca. Para conseguir esto procedemos a la

opción de aumentar el brillo y el contraste en GIMP:

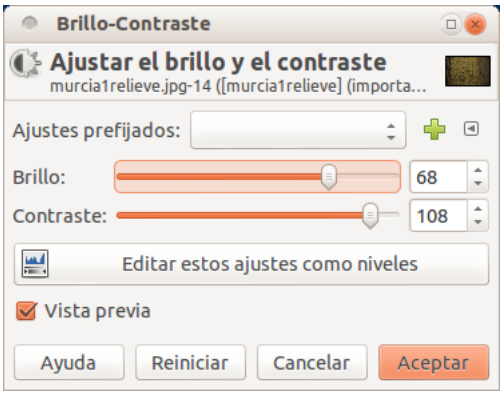


Figura 5.16: Ajuste de brillo y contraste

Mediante la opción **Colores->Brillo y contraste** de GIMP, estableceremos unos valores del brillo desplazando hacia la derecha (+x) el histograma. Asimismo expandiremos el histograma para aumentar el contraste. El resultado será el siguiente:

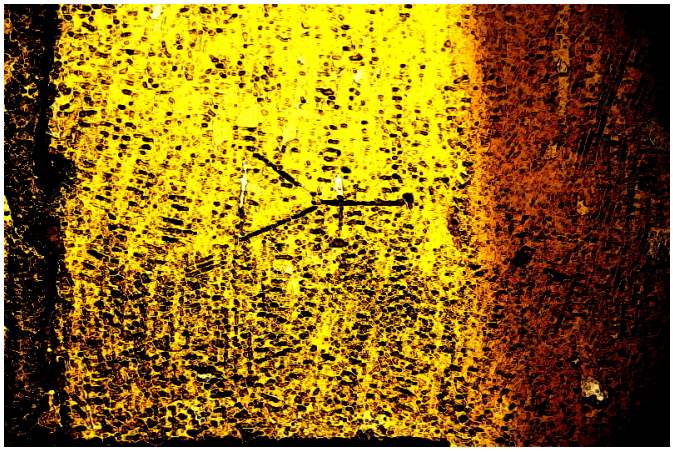


Figura 5.17: Resultado al aumentar el brillo y el contraste

Ahora podemos obtener el código de cadena más fácilmente como veremos a continuación.

5.4. CÓDIGOS DE CADENA

Una vez segmentada la imagen llegamos a la fase de descripción:

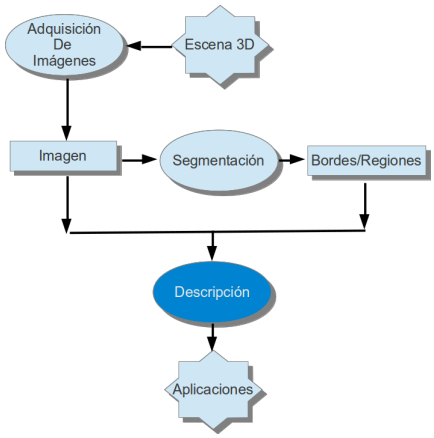


Figura 5.18: Fase de descripción

La idea ahora es realizar un procesamiento de la marca de cantero para obtener un código 8-direccional.

Para ello utilizaremos el algoritmo prototípico implementado en la herramienta ImageJ. Este algoritmo está basado en la técnica de *búsqueda en profundidad* para la búsqueda en profundidad en una imagen dividida en celdas de 10 x 10. Cuando el algoritmo encontraba una celda con una cantidad de color negro mayor que un umbral T, se seleccionaba como candidata y se mostraba su 8-dirección. De esta forma y si procedemos a ejecutar el programa mediante ImageJ, obtenemos el siguiente resultado: [4]

Al convertir la imagen en modo monocromático en el algoritmo de ImageJ se pierde parte de la información relacionada con zonas sensibles de la marca como pueden ser: zonas restauradas con yeso, zonas de poco relieve y/o visibilidad (iluminación), etc.

Obviamente, esta solución no es óptima, como veremos en la investigación, pues obtiene el código de cadena en uno solo de sus segmentos, por lo que para obtener el código completo sería necesario la obtención y la concatenación de todos los segmentos

[4] Como veremos en la sección de investigación, el algoritmo preliminar en ImageJ difiere en cuanto al algoritmo de 256 tonos de grises de Matlab.

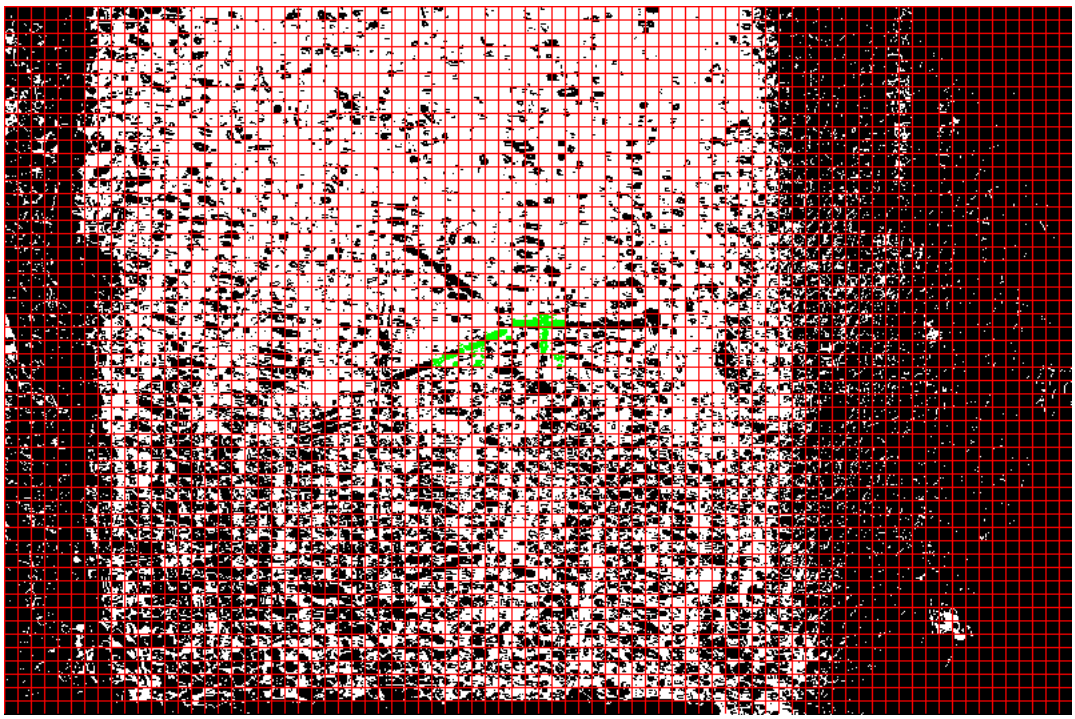


Figura 5.19: Obtención de código de cadena

[5]

De la no eficacia del algoritmo del código de cadena anterior, se desprende la necesidad de investigar por medio de métodos conexionistas, es decir, mediante Redes Neuronales que permitan una descripción mucho más potente.

Finalmente, la búsqueda de este algoritmo supondrá un verdadero desafío a la hora de investigar la solución.

[5] El código de cadena para este segmento es: E-S-S-SE-O-O-SO-O-SO-S-NO-SO-O.

6. INVESTIGACIÓN Y MODELOS

Comenzamos a abordar los diferentes modelos utilizados para la investigación, resultado de los objetivos y las técnicas científicas explicadas anteriormente. La problemática expuesta en el apartado anterior ha llevado la mayor parte de tiempo de trabajo y en algunas ocasiones el resultado no ha sido enteramente satisfactorio. Como veremos en el siguiente apartado, antes de comenzar con la descripción de las marcas de cantero se han aplicado unos modelos simbólicos de segmentación comunes a todas la imágenes, de cara a preparar mejor su análisis y procesamiento.

6.1. MODELOS SIMBÓLICOS COMUNES PARA LA SEGMENTACIÓN

Por modelos simbólicos entenderemos los métodos inspirados en algoritmos y simbología matemática y basados en estudios científicos para el análisis digital de la imagen. [6]

6.1.1. FORMATOS DE CAPTURA

La primera operación necesaria para trabajar con imágenes es poder adquirirlas desde algún formato gráfico. Para la realización de la investigación se han utilizado principalmente los formators JPEG y PNG que se pasan a describir a continuación:

1. **JPEG**: Son las siglas de *Joint Photographic Experts Group*. La compresión JPEG utiliza una compresión de tipo destructiva, por lo que desde el punto de vista de la calidad, esta compresión implica cierta pérdida de información dependiendo del porcentaje de compresión seleccionado. En general, JPEG descarta cierta información que considera redundante e innecesaria, es decir, el algoritmo pierde algunos de los datos, ya que identifica e ignora los píxeles que no son esenciales para la calidad global de la imagen; por ejemplo una zona continua de un mismo color. Las imágenes que presentan cambios bruscos de color no se comprimen bien. Es el formato más utilizado en imágenes en Internet, por lo que la mayoría de imágenes de prueba extraídas de este medio se encuentran en este formato. La compresión utilizada en las imágenes de prueba no ha superado el ratio 15:1, por lo que la calidad ha resultado muy satisfactoria.
2. **PNG**: Cuyas siglas son *Portable Network Graphics* utiliza una tecnología de compresión llamada “deflación” usada en muchos programas de software libre. PNG es una buena técnica para almacenar dibujos, texto y gráficos icónicos. Por este motivo se ha elegido como formato principal para grabar las marcas y los muros en escala de grises.

La función utilizada en Matlab para la lectura de ficheros ha sido:

[6] Véase [1]

```
Imagen = imread(fichero JPG|PNG)
```

6.1.2. ESCALA DE GRISES

Para la conversión de imágenes a tonos de grises se procede primeramente a convertir la imagen a grises con la herramienta GIMP. Concretamente con las opciones **Imagen->Modo->Escala de grises**. Es decir, que si tenemos una imagen con tres canales de información R,G y B, la imagen resultante de un sólo canal en escala de grises se calcula como:

$$color_{gris} = \frac{color_r + color_g + color_b}{3}$$

Figura 6.1: Ejemplo de imagen en tonos de grises

6.1.3. IMAGEN EN BLANCO Y NEGRO

Para convertir la imagen en blanco y negro es necesario realizar una transformación básica píxel a píxel aplicando un *operador umbral*:

$$f(x) = \begin{cases} 0 \text{ para } x \leq p_1 \\ 255 \text{ para } x > p_1 \end{cases} \tag{6.1}$$

En la ecuación 6.14 la imagen resultante f(x) es obtenida a partir de un valor de umbral definido como p_1. De esta forma obtenemos una imagen en blanco y negro como la mostrada a continuación:

Figura 6.2: Ejemplo de imagen en blanco y negro

El código Matlab utilizado para convertir una imagen RGB en tonos de grises a una en blanco y negro, es la siguiente:

```
BW = im2bw(I, level)
```

donde I debe ser una imagen en escala de grises y *level* el valor p_1.

6.1.4. TRANSFORMACIONES GEOMÉTRICAS

- **Escalado:** En la transformación geométrica del escalado se realiza un zoom de la imagen dependiendo del valor de los factores de escala S_x y S_y. Cuando el factor de escala toma valores entre 0 y 1 se produce una reducción de la imagen, mientras que cuando el factor de escala toma valores mayores que 1 se amplía. En general, la fórmula para escalar una imagen es la siguiente:

$$\begin{aligned} x &= S_x i \\ y &= S_y j \end{aligned}$$

o equivalentemente:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \tag{6.2}$$

Después de esta transformación geométrica es necesario realizar una *interpolación bicúbica* que es una de las mejores formas de interpolación.

La interpolación bicúbica tienen un núcleo de interpolación definido como:

$$h(x) = \begin{cases} 1 - 2|x|^2 + |x|^3 \text{ si } 0 < |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 \text{ si } 1 < |x| < 2 \\ 0 \text{ de otro modo} \end{cases} \tag{6.3}$$

- **Rotación:** La rotación se ha utilizado para mejorar la detección de marcas de cantero considerando las diversas posiciones de dibujo de la marca por el operario con respecto al patrón.

Las ecuaciones de rotación dado un ángulo θ son:

$$\begin{aligned} x &= cos\theta\, i - sen\theta\, j \\ y &= sen\theta\, i + cos\theta\, j \end{aligned}$$

o equivalentemente:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} cos\theta & -sen\theta & 0 \\ sen\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \tag{6.4}$$

Figura 6.3: Patrón original

Figura 6.4: Imagen rotada 90º

6.1.5. BRILLO Y CONTRASTE

Por medio del ajuste del histograma de la imagen en cuestión se puede ajustar los valores de brillo y contraste.

- **Desplazamiento del histograma**: El desplazamiento del histograma se usa para aclarar u oscurecer una imagen pero manteniendo la relación entre los valores de los niveles de gris. Esta operación se realiza por medio de adiciones o sustracciones de un número fijo a todos los valores de gris:

$$g(i,j) = f(i,j) + desplazamiento \tag{6.5}$$

En la ecuación 6.5 se considera que los valores que sobrepasan el máximo y el mínimo se redondean al máximo y el mínimo posibles permitidos.

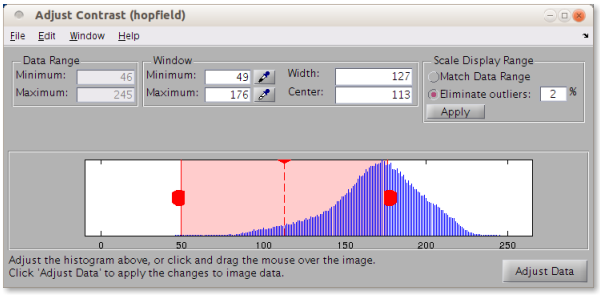


Figura 6.5: Histograma a desplazar a brillo

Figura 6.6: Imagen con brillo

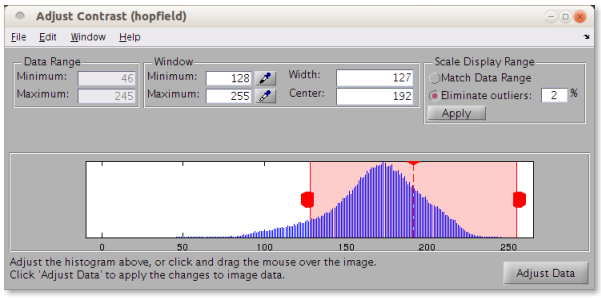


Figura 6.7: Histograma a desplazar a oscuro

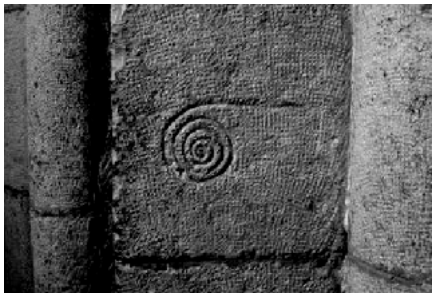


Figura 6.8: Imagen con oscuridad

- **Contracción del histograma**: Por medio de la contracción del histograma se consigue una disminución del contraste de la imagen. La función que la define es:

$$g(i,j) = \left[\frac{C_{MAX} - C_{MIN}}{f(i,j)_{MAX} - f(i,j)_{MIN}} \right] [f(i,j) - f(i,j)_{MIN}] + C_{MIN} \tag{6.6}$$

Donde $f(i,j)_{MIN}$ es el menor valor del nivel de gris de la imagen de entrada y $f(i,j)_{MAX}$ el mayor; C_{MAX} y C_{MIN} corresponden al máximo y mínimo valores deseados en la compresión del histograma.

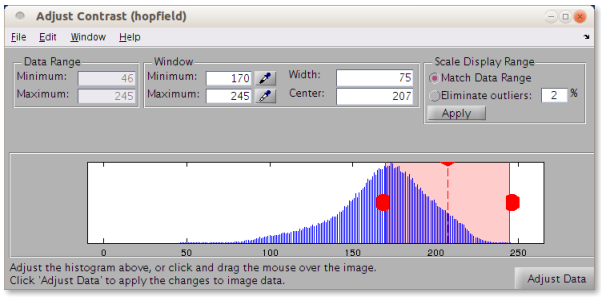


Figura 6.9: Histograma de disminución del contraste

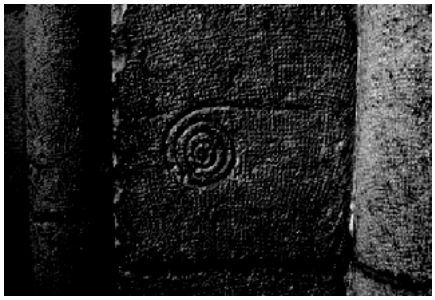


Figura 6.10: Imagen con contraste bajo

- **Expansión del histograma**: De forma contraria, la expansión del histograma consigue un aumento del contraste, tal como viene definida en la ecuación 6.7

$$g(i,j) = \left[\frac{f(i,j) - f(i,j)_{MIN}}{f(i,j)_{MAX} - f(i,j)_{MIN}} \right] [MAX - MIN] + MIN \tag{6.7}$$

Donde $f(i,j)_{MIN}$ es el menor valor del nivel de gris de la imagen de entrada y $f(i,j)_{MAX}$ el mayor; MAX y MIN corresponden al máximo y mínimo valores posibles de los niveles de gris (para una imagen de 8 bits sería 0 y 255).

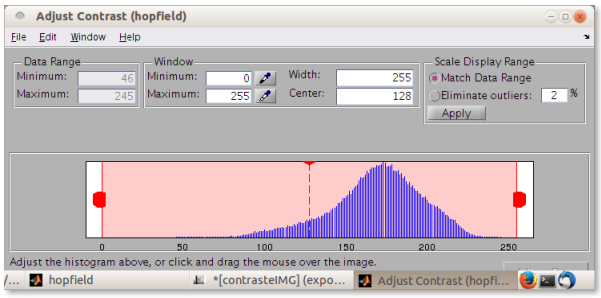


Figura 6.11: Histograma de aumento del contraste

Figura 6.12: Imagen con contraste alto

6.1.6. OPERACIONES MORFOLÓGICAS

Las operaciones morfológicas han permitido aplicar mejoras sobre la imagen en tonos de grises de la marca de cantero. A este respecto, dada una imagen en tonos de grises se han podido aplicar las siguientes primitivas:

- **Dilatación**: La operación morfológica de la dilatación se identifica con el símbolo $\oplus$ y combina la suma de vectores. La dilatación implica la siguiente operación:

$$X \oplus B = \{d \in E^2 : d = x + b \text{ para cada } x \in X \text{ y } b \in B\} \tag{6.8}$$

Y en Matlab se ha aplicado utilizando la siguiente secuencias de instrucciones:

```
se = strel('disk',1)
imdilate(imagen,se)
```

donde `strel` crea una estructura morfológica básica en forma de disco de radio 1. Esta estructura corresponde al valor B de la ecuación 6.8. Finalmente se llama a la función de Matlab `imdilate` con la imagen X en cuestión.

- **Erosión**: La transformación morfológica de la erosión se representa con el símbolo ⊗ y combina dos conjuntos utilizando la resta de vectores. La operación de erosión se especifica en la siguiente ecuación:

$$X \otimes B = \left\{ d \in E^2 : d + b \in X \text{ para cada } b \in B \right\} \tag{6.9}$$

  De esta forma en Matlab se utiliza llamando a la siguiente secuencia de instrucciones:

```
se = strel('disk',1)
imerode(imagen,se)
```

  que como se comentó en la dilatación, *se* correponde a una estructura morfológica en forma de disco que se aplica en la ecuación 6.9.

El motivo de utilizar operaciones de erosión y dilatación es con el fin de facilitar la mejor visión de la estructura geométrica y lineal de la marca de cantero, pues debido a su desgaste sobre la piedra, no se ven claros los contornos. Sin embargo, la operación de dilatación permite afinar a más contenido la imagen de la marca, mientras que la erosión permite eliminar formas indeseadas.

Figura 6.13: Erosión de espiral X1

Figura 6.14: Dilatación de espiral X1

6.1.7. EXTRACCIÓN DE BORDES

Una de las técnicas más importantes para preparar la imagen para el análisis por Hopfield y Correlación es la de extracción de bordes. En general se ha utilizado la técnica de *Sobel*. Los operadores de Sobel se basan cálculos de primera derivada, en los que se obtiene los valores G_x y G_y como:

$$G_x = \frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} \qquad G_y = \frac{f(y+\Delta y) - f(y-\Delta y)}{2\Delta y}$$

Los operadores del gradiente en general tienen el efecto de magnificar el ruido subyacente en la imagen. Sin embargo, los operadores de Sobel tienden a suavizar

la imagen, eliminando parte del ruido y por consiguiente, minimiza la aparición de falsos bordes debido al efecto de magnificación del ruido por parte de los operadores derivada:

Por tanto, los operadores de Sobel se definen como:

$$G_x = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \tag{6.10}$$
$$G_y = (z_7 + 2z_8 + z_9) - (z_1 + 2_z 2 + z_3) \tag{6.11}$$

y donde:

$$\begin{bmatrix} z_1 & z_2 & z_3 \\ z_4 & z_5 & z_6 \\ z_7 & z_8 & z_9 \end{bmatrix}$$

se obtiene que las máscaras de convolución de *Sobel* son:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Finalmente se marcará como punto de borde $g(x,y)$ si dado un umbral T y sabiendo que $|G| = \sqrt{G_x^2 + G_y^2}$:

$$g(x,y) = \begin{cases} 1 \text{ si } |G| > T \\ 0 \text{ si } |G| \leq T \end{cases} \tag{6.12}$$

Sobel se ha aplicado en los patrones de aprendizaje de *Redes de Hopfield* quedando los siguientes resultados:

- **Redes de Hopfield (resultado de aplicación Sobel):**

Figura 6.15: Patrón espiral

Figura 6.16: Patrón cruz

Figura 6.17: Patrón estrella

6.2. MODELOS CONEXIONISTAS PARA LA DESCRIPCIÓN

Por modelos conexionistas entenderemos los métodos basados en técnicas de Inteligencia Artificial Bioinspirada, es decir, las redes neuronales artificiales de aprendizaje supervisado y no supervisado. Estas redes están inspiradas en la naturaleza, de hecho, son muchos los autores que hacen un paralelismo entre los microprocesadores y el cerebro, de tal forma que se podría decir que en un futuro próximo el número de transistores sería equivalente al número de neuronas. En general la diferencia entre el procesamiento del cerebro y de un microprocesador radica en que el primero procesa la información de forma paralela y concurrente, mientras el segundo lo hace de forma secuencial.

Las *redes neuronales artificiales o RNAs* intentan emular el procesamiento de del cerebro; aunque todavía estamos lejos de conseguirlo. Su elemento constructivo básico es el perceptrón que en cierta forma está inspirada en de la neurona biológica:

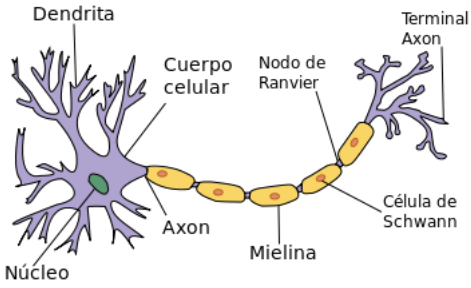


Figura 6.18: Neurona. Tomada de US Federal (public domain).

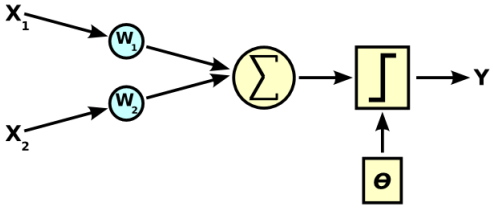


Figura 6.19: Perceptrón. Creative Commons.

Las entradas del perceptrón son las *dentritras*, es decir, $x_1, x_2, ..., x_n$ y donde el *axón* es la salida Y. Cada una de las entradas se les asignará un peso determinado llamado $w_1, w_2, ..., w_n$. La variable θ indica el umbral de disparo. El valor de la salida Y estará comprendida entre 0 y 1.

De esta forma:

$$Y = \begin{cases} 1 \text{ si } \sum_i x_i \cdot w_i > \theta \\ 0 \text{ si } \sum_i x_i \cdot w_i \leq \theta \end{cases} \tag{6.13}$$

La tarea ahora es ir ajustando los valores de los pesos a través de un proceso de *entrenamiento* (aprendizaje supervisado) de forma que se va proveyendo a la red de ejemplos y se va ajustando la respuesta en función de estos. Se suele añadir un

factor más llamado factor de aprendizaje λ que indica a qué velocidad aprende la red. En general, suele ser habitual un valor λ de 0.2.

Por lo tanto durante el proceso calcularemos el *error* de obtenido para reajustar los pesos y el umbral. De esta forma, y suponiendo que la salida deseada sea z, se calcula como:

$$error = (z - Y) \qquad \text{cuando } z \neq Y$$

De ahí obtenemos el nuevo incremento al umbral de disparo:

$$\Delta\theta = -(\lambda \cdot error)$$

y de igual forma se ajustan los pesos:

$$\Delta w_i = -(\lambda \cdot error \cdot x_i)$$

Muchos de los problemas resueltos con el método de perceptrón son linealmente separables, aunque es muy frecuente encontrarse con otros tipos de problemas, como la función XOR, cuyo plano no es linealmente separable.

El modelo de perceptrón presentado aquí responde a una red neuronal de una sola capa y una única neurona. El la sección 6.2.2 trataremos más en profundidad las redes neuronales multicapa que se han utilizado para la descripción de marcas de cantero.

La taxonomía de las redes neuronales viene determinada por el siguiente esquema:

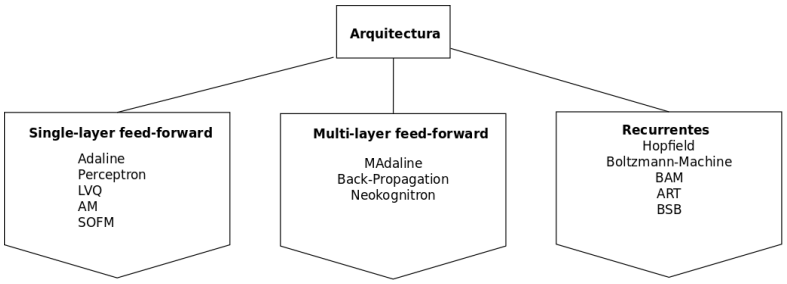


Figura 6.20: Taxonomía redes neuronales

Las redes utilizadas en la investigación han sido del tipo *Multi-layer feed-forward Back-Propagation* y *Hopfield*.

6.2.1. REDES DE HOPFIELD

Las redes de Hopfield, —llamadas así en honor a su inventor John Hopfield—, son un tipo de red neuronal basadas en aprendizaje no supervisado de tipo hebbiano, es decir, que a diferencia de las redes neuronales del tipo back-propagation feed-forward, no necesitan de un entrenamiento previo para especificar cuál es la respuesta correcta para un conjunto de patrones.

En general, las redes de Hopfield recrean con más precisión el modelo natural del cerebro humano e imita en cierto modo la forma de procesar del neocórtex, que parece almacenar los recuerdos usando una memoria asociativa. Por ejemplo, escuchar una canción de nuestra infancia puede evocar un recuerdo que permanece latente en nuestro inconsciente. Estos recuerdos pueden no estar muy claros o no ser muy exactos, de tal manera que, aunque permanezcan distorsionados, el cerebro tiende a recomponerlos. Es como si distinguiéramos a una persona a la que no hemos visto durante años por alguno de sus rasgos actuales.

Las redes de Hopfield tienen una gran aplicación para el reconocimiento óptico de caracteres (OCR) y para los captcha (sistemas que evitan que agentes software suplanten a humanos en formularios de registro Webs).

La arquitectura de las redes de hopfield se basa en una red monocapa con N neuronas cuyos valores de salida son binarios: 0/1 ó -1/+1. Se puede modelar una red de hopfield como un grafo no dirigido y completamente conectado, en donde cada nodo es una neurona sin una relación reflexiva y las aristas entre neuronas los pesos de entrada a cada una de ellas.

Las salidas (para un caso típico de implementación) se calculan como:

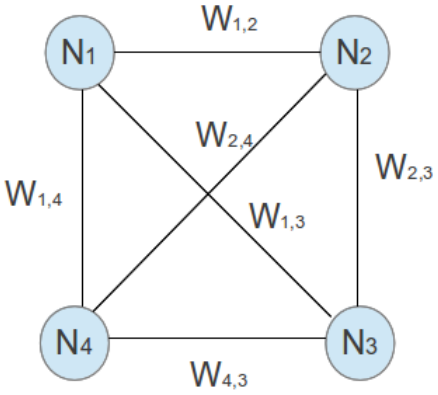


Figura 6.21: Ejemplo de red de Hopfield

$$a_i = \begin{cases} 1 & si \quad \sum_j w_{ij} s_j > \phi_i \\ -1 & \text{en otro caso} \end{cases}$$

y donde:

- w_{ij} es el peso de la conexión de la neurona j con la i. $w_{ii} = 0$ y $w_{ji} = w_{ij}$
- s_j es el estado de la neurona j.
- θ es el umbral de activación de la neurona i.

de esta forma la función de activación se representa como:

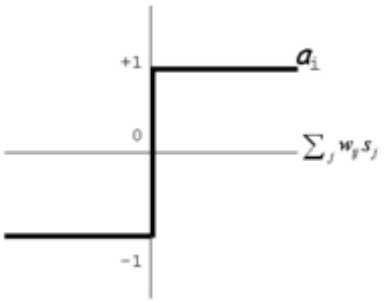


Figura 6.22: Función de activación de una red de Hopfield

Hopfield demostró que se podía definir una función de energía para cada posible patrón de activación o estado de la red:

$$Energía = -\frac{1}{2} \sum_{i,j} w_{ij} s_i s_j + \sum_i \theta_i \, s_i$$

donde este valor tiende a converger a un mínimo local para un estado estable de la red.

Aplicación del algoritmo en casos reales de marcas de cantero

Para la aplicación práctica de las redes de Hopfield en un algoritmo que permita trabajar con marcas de cantero se supone en primer lugar la existencia de tres imágenes de entrenamiento:

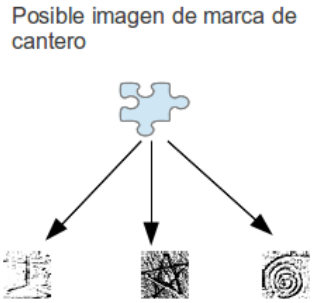


Figura 6.23: Hipótesis de base

De esta forma es necesario leer primeramente las imágenes de entrenamiento:

```
% imagenes de entrenamiento
a = im2bw(imread('../muestras/cruz_hop.png'),0.8)
b = im2bw(imread('../muestras/espiral_hop.png'),0.7)
c = im2bw(imread('../muestras/estrella_hop.png'),0.3)
```

donde primeramente se procede a leerlas de fichero en formato PNG y posteriormente se binarizan a un determinado umbral [7].

Puesto que las imágenes tienen una dimensión de 32 x 32, el siguiente paso es convertirlas a un vector lineal de 1024 elementos mediante la función **avector**:

```
function x = avector(a, n)
   k=0;
   x = [];
    for i=1:n
       for j=1:n
           k = k + 1;
           x(k) = a(i,j);
       end
    end
  end
```

Por lo tanto el código continuará como:

```
aP = avector(a,32);
bP = avector(b,32);
cP = avector(c,32);
```

el siguiente paso es convertir los valores 0 y 1 a valores -1 y +1 respectivamente:

```
e1 = 2*aP - 1;
e2 = 2*bP - 1;
e3 = 2*cP - 1;
```

ahora procedemos a crear la matriz de pesos:

```
% Crea matriz identidad
id = eye(N);

t1 = e1' * e1 - id;
t2 = e2' * e2 - id;
t3 = e3' * e3 - id;
```

[7] Como comentábamos en la sección 6.1.7 estas imágenes se les han extraído los bordes mediante Sobel

```
t = t1 + t2 + t3;
```

por último obtenemos la imagen *target* a probar **Vx**:

Listing 1: listado1

```
1
2       vXSillar = vX;
3
4       alfa = 0; % Comienza en 0 grados
5
6       vXRot = vX;
7
8       while (alfa < 360)
9
10           if (alfa >= 90)
11           % Convierte a vector horizontal la ventana
12
13                 vXRot = imrotate(vXSillar, alfa);
14           end
15
16          % Escala ventana
17          vXCap = imresize(vXRot, [32 32], 'bicubic', 'antialiasing',true);drawnow;
18
19          vXSim = avector(im2bw(vXCap,0.6),32);
20
21          vXSim = 2*vXSim - 1;
22
23          fprintf('Comparando...\n');
24          [i,sx] = hopfield(t,vXSim,N);
25          fprintf('Iteraciones = %d\n', i);
26          resultado(sx, e1, 'CRUZ\n');
27
28
29          fprintf('Comparando con ESPIRAL\n');
30          [i,sx] = hopfield(t,vxSim,N);
31          fprintf('Iteraciones = %d\n', i);
32          resultado(sx, e2, 'ESPIRAL\n');
33
34          fprintf('Comparando con ESTRELLA\n');
35          [i,sx] = hopfield(t,vxSim,N);
36          fprintf('Iteraciones = %d\n', i);
37          resultado(sx, e3, 'ESTRELLA\n');
38
39          alfa = alfa + 90;
40      end
```

La imagen se rota en las cuatro direcciones de los puntos cardinales en sentido contrario a las agujas del reloj, es decir, 0º, 90º, 180º y 270º. De esta forma es posible detectar las posibles orientaciones en las que el cantero pudo dibujar la misma marca.


Figura 6.24: 0º


Figura 6.25: 90º


Figura 6.26: 180º


Figura 6.27: 270º

En las figuras superiores se puede observar la cruz de la Catedral de Murcia rotada en las cuatro direcciones.

*El problema surge en las líneas **17** y **19** del listado 1 puesto que al reducir la imagen al tamaño de 32 x 32 con la función **imresize** y al convertirla a binaria con un determinado umbral **im2bw(img, umbral)** la calidad de la muestra se reduce en consideración como se puede apreciar en las figuras anteriores.*

Por último es necesario aplicar el algoritmo de Hopfield a la muestra (target) obtenida mediante la llamada a la función **Hopfield**:

Listing 2: listado2

```
1  function [i sx] = hopfield(t, s_ant, N)
2      i = 1;
3      iguales = false;
4      while (i <= 5 && ~iguales) % Numero de iteraciones
5            i
6            m = s_ant*t;
7            sx = escaloniza(m,s_ant, N);
8            energia = 0;
9            for j=1:N
10               for k=1:N
11                   energia = energia + -1/2*t(j,k)*sx(j)*sx(k);
12               end
13           end
14
15           energia
16
17           iguales = isequal(sx,s_ant);
18           s_ant = sx;
19           i = i + 1;
20     end
21 end
```

Obviamente en las pruebas esta función se llama con un valor N = 1024. El procedimiento de Hopfield consiste en multiplicar la matriz de pesos por la matriz target convertida en vector de 1024 elementos. El proceso se detiene bien porque resulta igual a la matriz de entrada o porque se llega a un máximo de 5 iteraciones predefinidas. La función **escaloniza** realiza la siguiente operación matemática:

$$f(x) = \begin{cases} -1 \text{ si } m < 0 \\ +1 \text{ si } m > 0 \\ s\_ant_k \text{ si } m = 0 \end{cases} \tag{6.14}$$

Puesto que la función de Hopfield devolverá un vector de 1024 elementos con el resultado de la imagen de entrenamiento a la que mejor converge, es posible que algunas veces termine con un resultado *híbrido*, es decir, que se encuentre como una mezcla de dos imágenes de entrenamiento. Para prevenir esta situación se ha definido una función de semejanza comparando la imagen de resultado con la imagen original de entrenamiento a partir de un nivel o umbral de porcentaje de coincidencia.

Lo que básicamente hace esta función **parecido** es recorrer los dos vectores(imagen de entrenamiento y resultado) y para cada coincidencia binaria se incrementa +1 el contador de coincidencias. De esta forma el porcentaje de correspondencia se determina como:

$$porcentaje\ de\ coincidencia = \frac{cont}{1024} * 100$$

donde *cont* es el número de veces que son iguales ambos vectores.

Finalmente si ambos vectores superan el umbral se procede a representarlo gráficamente mediante la conversión a 2D:

Listing 3: listado3

```
function b = resultado(sx, target, tolerancia)
    if (parecido(sx,target) > tolerancia)
        r = [];
        for i=1:32
            for j=1:32
                    r(i,j) = sx((i-1)*32 + j);
                if (r(i,j) == -1)
                    r(i,j) = 0;
                end
            end
        end
          b = true;
    else
          b = false;
    end
end
```

Aplicación de demostración

Para aplicar el algoritmo anterior se ha procedido a crear una aplicación de demostración con interfaz gráfica de usuario que lea una imagen de archivo con una marca de cantero y determine a cuál de tres imágenes de entrenamiento corresponde. Para afinar la aplicación del algoritmo se acompaña la interfaz con un conjunto de operaciones simbólicas tratadas en el apartado 6.1.

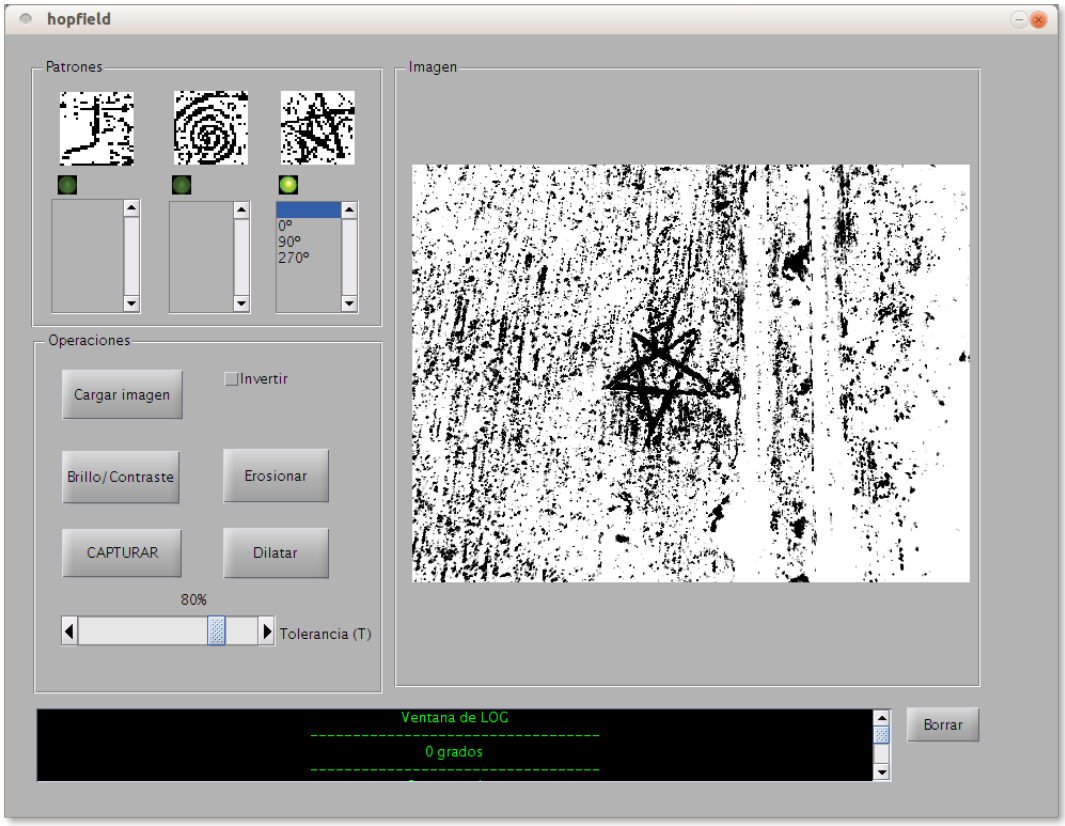


Figura 6.28: Aplicación de demostración de Hopfield

6.2.2. Redes Neuronales

En esta sección se tratarán las redes neuronales multicapa como solución de aprendizaje y descripción de marcas de cantero más complejas. A diferencia de lo que se explicaba al comienzo de sección para el perceptrón simple, las redes multicapa nos permiten resolver problemas *no linealmente separables*.

Básicamente la estructura de una red neuronal multicapa está formada por un conjunto de neuronas de entrada, ocultas y de salida, tal como se muestra en la siguiente imagen:

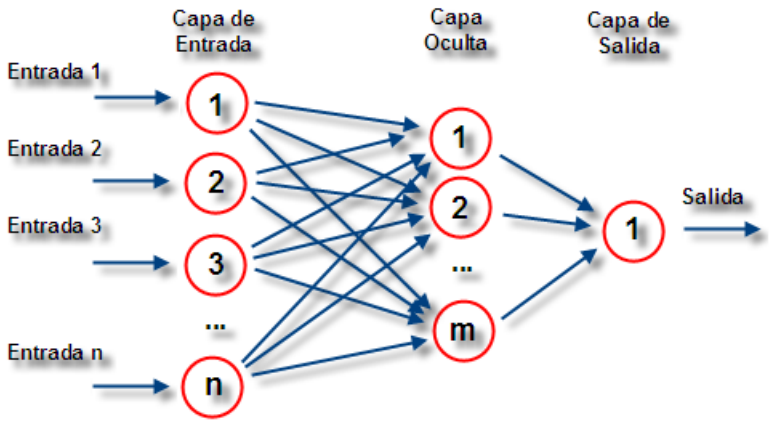


Figura 6.29: Ejemplo de red neuronal biclase. GNU.

Se puede observar como cada una de las neuronas en cada capa se conecta a todas la neuronas de la capa anterior a través de un valor $W_{i,j}$ que llamamos peso y siguen la misma filosofía que en el perceptrón simple.

En general, sin capas ocultas se pueden resolver problemas linealmente separables, mientras que con un única capa oculta se puede resolver problemas mediante una curva, y, con dos o más, problemas con separaciones arbitrarias.

Con el perceptrón simple se consideraba la activación de la neurona si la suma de los pesos multiplicados por las entradas era mayor que el umbral determinado θ. No obstante, para las redes multicapa se necesita una función de activación diferente, derivable, que la denominaremos función *sigmoide*:

$$f(x) = \frac{1}{1+e^x} \tag{6.15}$$

En la práctica de la aplicación de demostración implementada para las redes neuronales es posible utilizar otras diferentes en el menú de configuración de la red neuronal de Matlab.

En la ecuación 6.15 es importante destacar que los valores resultado están acotados entre -1 y +1 y crece rápidamente hasta 1 para valores mayores que cero y decrece igualmente hasta -1 para valores menores que cero. En la siguiente imagen se puede apreciar la representación gráfica de la función sigmoidea:

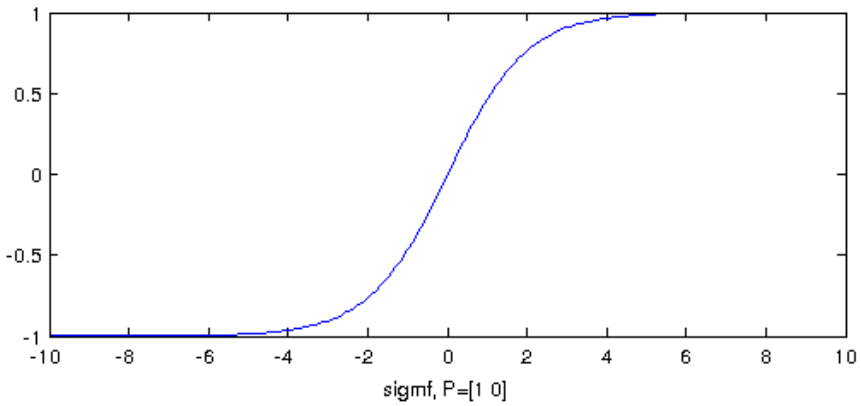


Figura 6.30: Función sigmoidea ploteada con Matlab. Tangente hiperbólica.

De esta forma las neuronas de la red se activarán siempre y cuando se cumpla la condición aplicada 6.15:

$$Y = f(\sum_i x_i \times w_i) \tag{6.16}$$

Se considera que un valor se activa cuando se aproxima a 1, por ejemplo el valor 0.9521.

En la capa de salida, el error se calcula de igual forma que para el perceptrón:

$$error = (z - Y)$$

y el incremento corrector como:

$$\Delta_i = f'(x_i) \times error$$

donde i se refiere al valor de la salida de la neurona i. Por lo tanto el error se debe propagar ahora hacia las capas ocultas como:

$$error = \sum_{i,j} \Delta_j \times w_{i,j}$$

Donde j es la salida de la neurona j.

A partir del error obtenemos el incremento de ajuste para la capa oculta i como:

$$\Delta_i = f'(x_i) \times error$$

Ahora es necesario ajustar los pesos realizando el siguiente intercambio:

$$intercambio = \Delta_{ni} \times x_{nj}$$

y donde resulta que:

$$w_{i,j} = w_{i,j} + (\lambda \times intercambio)$$

siendo n el número de la capa de salida u oculta y λ el factor de aprendizaje.

Método simbólico utilizado para la detección de sillares

Se ha utilizado un algoritmo iterativo para la detección de juntas de sillares en la aplicación de demostración de la sección de redes neuronales. Dicho algoritmo consiste básicamente en analizar una captura, fila por fila y columna por columna, en una imagen perpendicular a la cámara.

Es decir dada la siguiente imagen previamente binarizada a un determinado umbral:

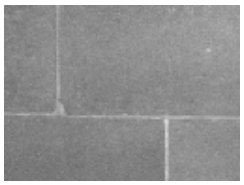


Figura 6.31: Sillar original

Figura 6.32: Sillar binarizado

Se trata de hacer un "escaneo" lineal vertical y horizontal de la siguiente forma:

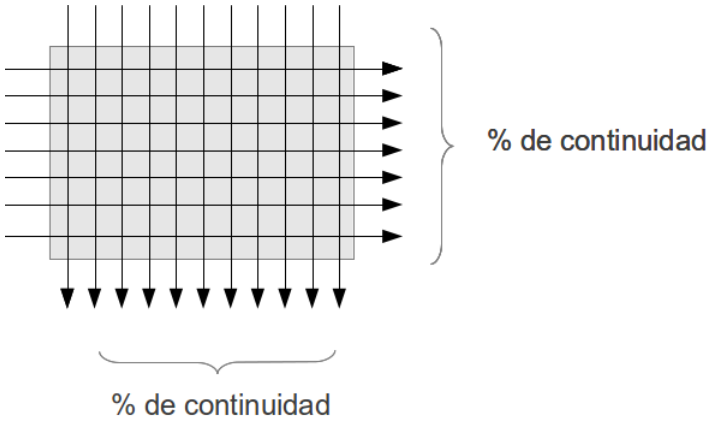


Figura 6.33: Escaneo del sillar a un determinado porcentaje de continuidad

Así para un determinado color (blanco o negro) se procede a aplicar el siguiente algoritmo:

Listing 4: listado4

```
1
2
3  % Algoritmo para deteccion de juntas de sillares
4
5  function [x y] = essillar(img)
6
7      [f c] = size(img) % Obtiene tamanio imagen
8      hm = img;
9
10     % PARTE HORIZONTAL
11
12     vMax = [];
13     r = 0;
14     for j=1:c % Por cada columna de la matriz de la imagen
15         k = 0;
16         roto = 0;
17         ant = hm(1,j); % Obtiene primer valor columna
18         max = 0;
19         for i=1:f % Recorre las filas
20             if ((hm(i,j) == 0) && (ant == 0)) % Si es color negro RGB(0,0,0). Minimo dos pixeles
21            % de continuidad
22                 k = k + 1;
23                 if (k > max) % Obtiene maximo de la fila que se mantiene continuo
24                     max = k;
25                 end;
26             else
27                 k = 1; % Determina una ruptura en la continuidad de la junta
28                 roto = roto + 1;
29             end
30         ant = hm(i,j); % Actualiza valor a analizar en posicion anterior
31         end
32        vMax = [vMax,max]; % Inserta en lista de maximos
33     end
34     x = vMax;  % Devuelve el maximo de continuidad de junta
35
36     % PARTE VERTICAL
37
38     vMax = [];
39     r = 0;
40     for i=1:f % Por cada fila de la matriz de la imagen
41         k = 0;
42         roto = 0;
43         ant = hm(i,1);
44         max = 0;
45         for j=1:c % Recorre las columnas
46             if ((hm(i,j) == 0) && (ant == 0)) % Si es color negro RGB(0,0,0). Minimo dos pixeles
47             % de continuidad
48                 k = k + 1;
49                 if (k > max)  % Obtiene maximo de la fila que se mantiene continuo
50                     max = k;
51                 end;
52             else
53                 k = 1; % Determina una ruptura en la continuidad de la junta
54                 roto = roto + 1;
55             end
56         ant = hm(i,j);
```

```
57          end
58          vMax = [vMax,max]; % Inserta en lista de maximos
59      end
60      y = vMax;
61  end
```

La idea del algoritmo es ir recorriendo cada fila y columna mientras exista un valor de continuidad en píxeles de blanco o de negro. Este valor de continuidad de píxeles se especifica con un porcentaje. Por ejemplo, si observamos la imagen 6.32 sabemos de antemano que debe existir al menos un mínimo de un 70 % de continuidad de blanco. Sin embargo, la medida para el color negro no es válida para detectar juntas de sillares, aunque podría tener cierto valor discriminador debido a su distribución uniforme.

Aplicación de demostración de red neuronal biclase

La aplicación creada para la descripción de marcas de cantero mediante una red neuronal biclase permite determinar si una determinada captura de un muro de una Catedral es o no una marca de cantero, es decir, el resultado de la capa de salida será afirmativo o negativo.

El primer requisito para comenzar a analizar una marca es cargar una imagen en formato de PNG o JPG en tonos de grises. A patir de entonces podemos configurar la red neuronal pulsando el botón *"Configurar"* donde nos aparecerá una pantalla que nos permite elegir el tipo de función de entrenamiento, de las cuales se puede elegir una de las siguientes:

Tipo	Descripción
trainscg	Scaled conjugate gradient backpropagation
trainb	Batch training with weight and bias learning rules
trainbfg	BFGS quasi-Newton backpropagation
trainbr	Bayesian regularization
trainc	Cyclical order incremental training w/learning functions
traincgb	Powell -Beale conjugate gradient backpropagation
traincgf	Fletcher-Powell conjugate gradient backpropagation
traincgp	Polak-Ribiere conjugate gradient backpropagation
traingd	Gradient descent backpropagation
traingdm	Gradient descent with momentum backpropagation
traingda	Gradient descent with adaptive lr backpropagation
traingdx	Gradient descent w/momentum and adaptive lr backpropagation
trainlm	Levenberg-Marquardt backpropagation
trainoss	One step secant backpropagation
trainr	Random order incremental training w/learning functions
trainrp	Resilient backpropagation (Rprop)
trains	Sequential order incremental training w/learning functions

Cuadro 6.1: Tabla de funciones de entrenamiento

Por lo general utilizaremos una función del tipo *trainscg* (la primera) que es un función basada en el *Gradiente conjugado y escalado* para redes backpropagation.

Después de darle a aceptar y en una pantalla posterior se nos presenta la lista de opciones de entrenamiento de la red:

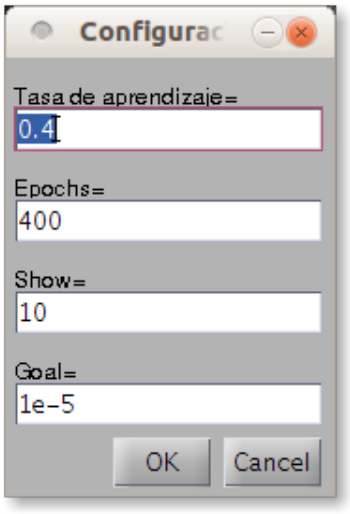


Figura 6.34: Opciones para el entrenamiento

donde cada opción representa la siguiente:

- **Tasa de aprendizaje**: Es lo que denominábamos la tasa de aprendizaje λ.
- **Epochs**: Son las iteraciones hasta conseguir la convergencia. Generalmente se utilizarán entre las 400 y las 4000.
- **Show**: Esta opción indica que cada 10 iteraciones el resultado del aprendizaje se mostrará en la pantalla.
- **Goal**: Indica que las iteraciones pararán cuando el valor de la función de rendimiento (función objetivo) caiga por debajo de este valor. Generalmente utilizaremos un valor menor o igual a 10^{-5} para un mejor rendimiento de la red.

Una vez configuradas la función de entrenamiento y las opciones adicionales podemos proceder a crear y entrenar la red mediante las funciones de Matlab:

```
1
2  % Entradas con vectores traspuestos
3
4  inputs  = [aP',bP',cP',dP',eP',fP', gP', hP',iP',jP', kP',lP',mP',nP',oP',pP',qP'];
5
6
7  % Crea red neuronal con 1024 entradas y una salida
8
9
10 net = feedforwardnet(1);
11
12 % Funcion de transferencia e iteraciones
13
14 net.trainFcn = 'trainscg';
15 net.trainParam.lr = 0.4;
16 net.trainParam.epochs = 400;
17 net.trainParam.show = 10;
18 net.trainParam.goal = 1e-5;
19
20 % Pesos (0.9 = Es marca de cantero; -0.9 = No es marca de cantero)
21
22 targets = [0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,-0.9,-0.9,
23 -0.9,-0.9,-0.9,-0.9,-0.9,-0.9,-0.9];
24
25
26 % Entrena la red neuronal
27 for i=1:20
28
29  net = train(net,inputs,targets);
30
31 end
```

Como se puede apreciar en el anterior listado, las entradas a la red deben ser vectores verticales (transpuestos) de 1024 elementos basados en una imagen de 32 x 32. De ser de una dimensión mayor la complejidad algorítmica crecería exponencialmente hasta bloquear el procesamiento.

Por otro lado hay que proporcionar los pesos de los patrones que se consideran marcas de cantero y los que no. Por ejemplo, en el listado anterior, los patrones de aP' a iP' se les establece un peso de 0.9; mientras que al resto de patrones de entrenamiento, comúnmente definidos como ruido de piedra y sillares, se les asigna un peso de -0.9.

El resultado de la ejecución de las anteriores líneas nos proporciona la siguiente visualización de la red neuronal:

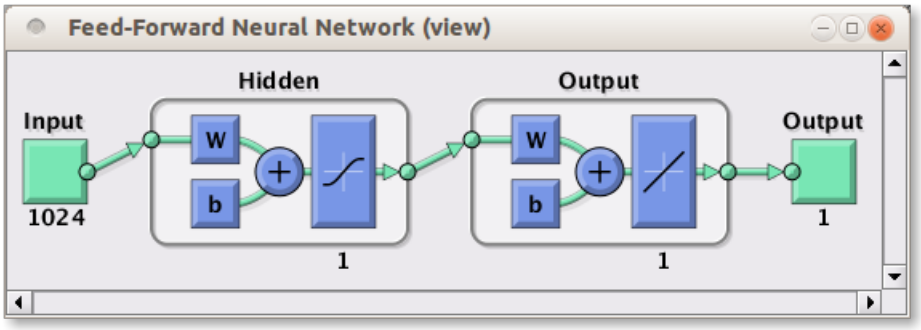


Figura 6.35: Red neuronal bi-clase creada

Donde encontramos una capa de entrada con 1024 entradas (32x32 píxeles), una capa oculta con 1 neurona y una capa de salida de discriminación con 1 neurona.

En la primera capa oculta tenemos una función de transferencia del tipo *Tangente Sigmoide* mientras que en la segunda capa (salida) se aplica una función de transferencia del tipo *Lineal*.

Durante el entrenamiento obtendremos la siguiente salida mostrada en la imagen 6.36:

Donde de ha utilizado un función MSE-Mean Squared Error (Error cuadrático medio) para calcular el error. De esta forma si suponemos una sola neurona en la capa de salida se calcula como:

$$MSE = \frac{1}{N_p} \sum_{i=1}^{N_p} \left({O_i}^a - {O_i}^d \right)^2 \tag{6.17}$$

y donde:

- N_p es el número de patrones de entrenamiento.
- ${O_i}^a$ es la salida actual de la red para el patrón de entrada i-ésimo.
- ${O_i}^d$ es la salida deseada el i-ésimo patrón de entrada.

Por lo tanto, después del procesamiento de entrenamiento podemos obtener los siguientes rendimientos según el MSE y los epochs.

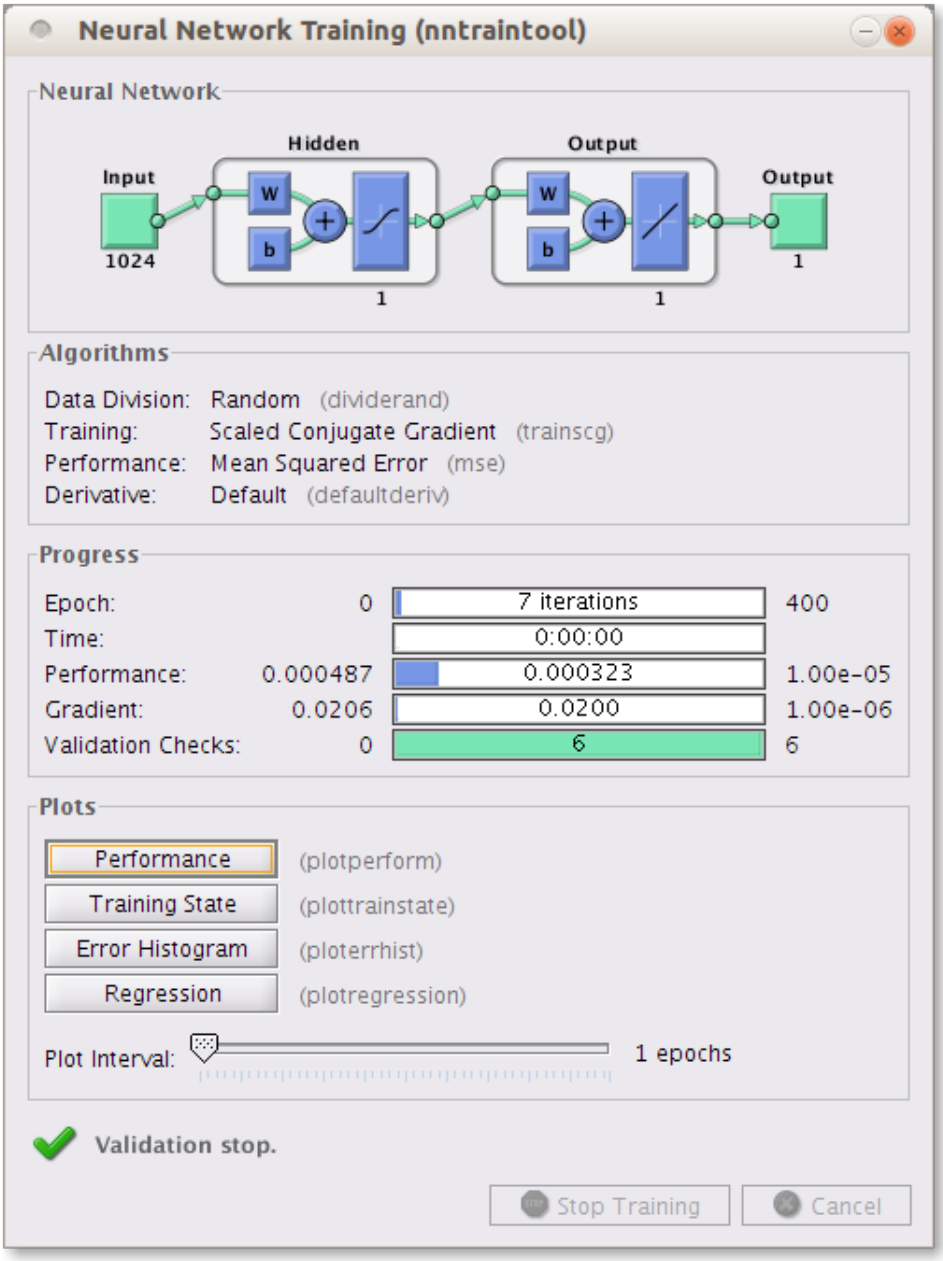


Figura 6.36: Proceso de entrenamiento

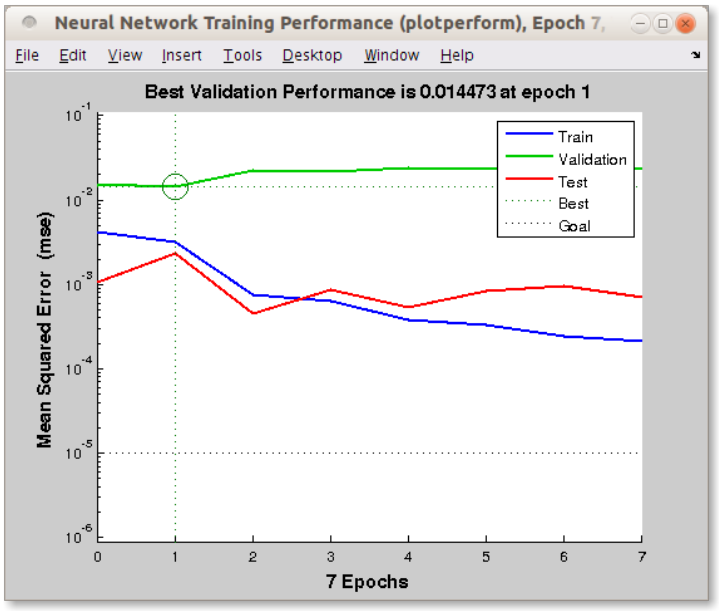


Figura 6.37: Calidad de entrenamiento baja

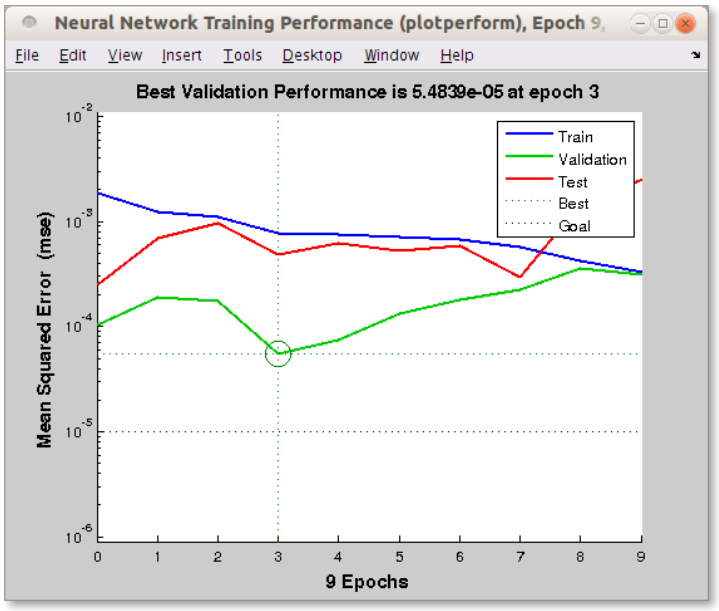


Figura 6.38: Calidad de entrenamiento alta

Por último solo queda capturar una zona del muro y establecer un nivel de tolerancia que oscilará entre 0 y 1.

En la imagen 6.39 de ejemplo se muestra la captura y detección de una marca de cantero (rectángulo verde) con un valor mínimo de T = 0.6 y la detección de junta de sillar (rectángulo blanco).

La simulación de la red neuronal se aplica en Matlab de la siguiente forma:

```
[Y,Pf,Af] = sim(net,vXSim');
 if (Y >= T)
```

Donde vXSim' (transpuesta) representa el vector de 1024 píxeles de una imagen rotada X4 y escalada; mientras que Y es un valor decimal comprendido entre [-0.9,0.9] que contiene el peso asignado al reconocimiento.

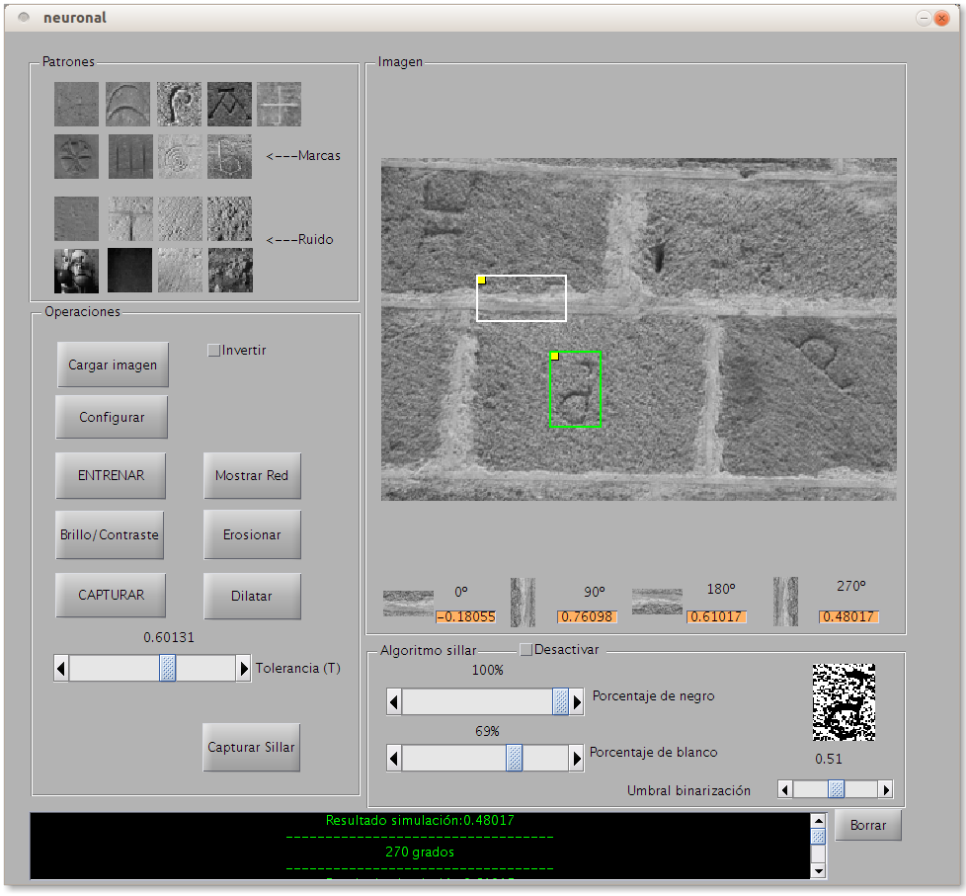


Figura 6.39: Aplicación de demostración

Aplicación de demostración de red neuronal multiclase

La aplicación de las redes neuronales multiclase a la detección de un conjunto discreto de marcas de cantero no difiere en gran medida de las anteriormente explicadas. Tan sólo requieren un cambio en la creación de la red y la configuración de las entradas y los targets.

De esta forma la aplicación trata de de discriminar una captura de un conjunto de 6 marcas predefinidas:

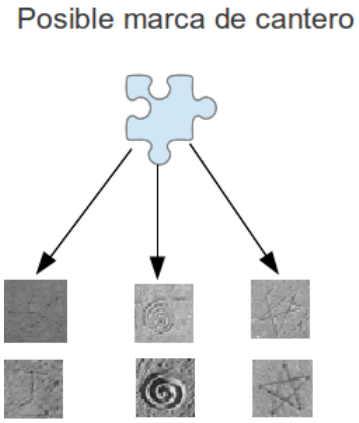


Figura 6.40: Detección en conjunto de 6 marcas posibles

Para ello será necesario configurar la red neuronal para 3 salidas.

```
inputs  = [a1P,a2P,b1P,b2P,c1P,c2P];

a = [+1 −1 −1]';
b = [−1 +1 −1]';
c = [−1 −1 +1]';

targets = [repmat(a,1,length(a1P)) repmat(a,1,length(a2P)) repmat(b,1,length(b1P))
 repmat(b,1,length(b2P)) repmat(c,1,length(c1P)) repmat(c,1,length(c2P))];
```

Donde *inputs* son las 6 imágenes de entrenamiento en vectores de 1024 elementos y *a, b, c* son los vectores transpuestos con las asignaciones de pesos para cada clase diferente. Es decir, la clase A para la marca de la “Cruz”, la clase B para la marca de la “Espiral” y la clase C para la marca de la “Estrella”. Finalmente se especifica el *target* replicando los patrones “a”, “b” y “c” tantas veces como indique la longitud de su correspondiente imagen de entrenamiento que será 1024.

A continuación será necesario crear la red neuronal con 3 neuronas de salida, mediante la siguiente sintaxis de instrucciones en Matlab:

```
net = feedforwardnet(3); %Aqui se crea la red con 3 neuronas en la salida

net = init(net);

net.trainFcn = 'trainscg';
net.trainParam.lr = 0.4;
net.trainParam.epochs = 4000;
net.trainParam.show = 10;
net.trainParam.goal = 1e-5;
```

Por último será necesario entrenarla:

```
for i=1:20
    [net,tr,Y,E] = train(net,inputs,targets);
end
```

Por lo que la red resultado será la siguiente:

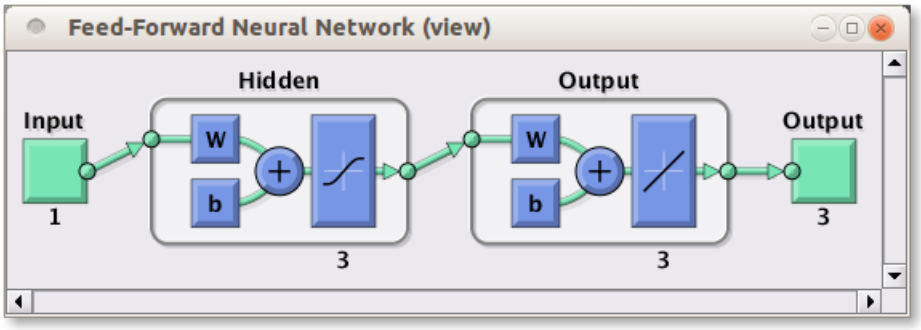


Figura 6.41: Red multiclase con 3 neuronas en la salida

Como se aprecia en la figura 6.41, la red neuronal proporciona una entrada con un vector de 1024 elementos no transpuestos,es decir, en serie, con una capa oculta consistente en una función de activación del tipo *Tangente Sigmoide* y una capa de salida con un función del tipo *Lineal*.

Como se puede apreciar en la imagen 6.42, la red neuronal detecta una salida de marca de cantero (Pentagrama) en los cuatro ángulos establecidos:

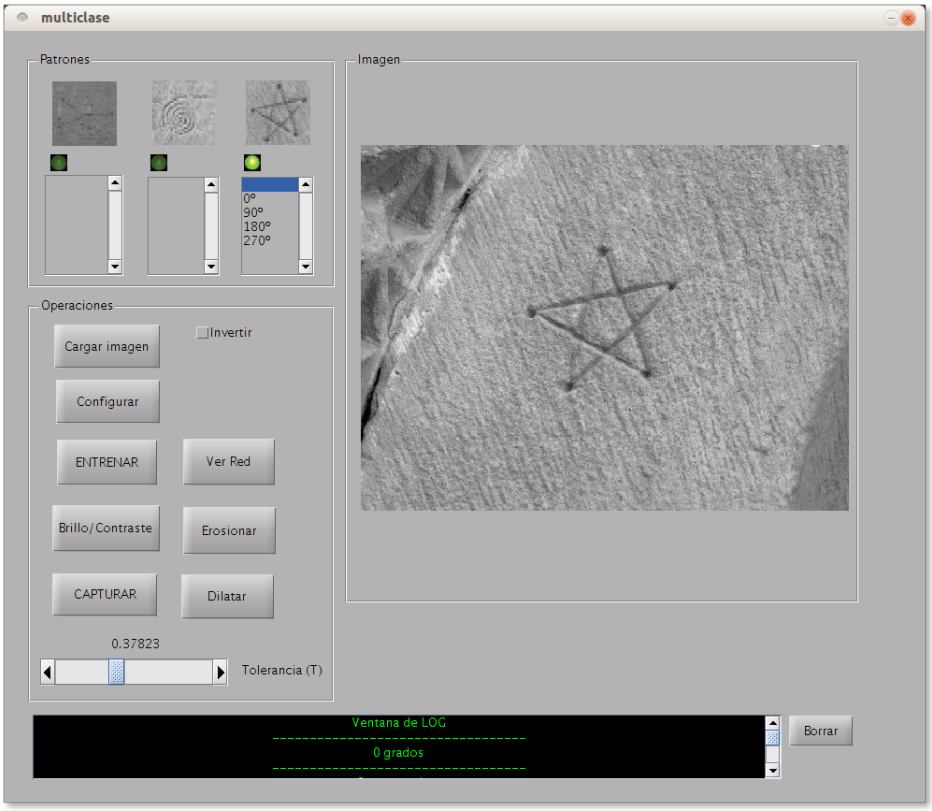


Figura 6.42: Detección de la clase C

Mediante la llamada a la función de Matlab que procesa 3 vectores de 1024 elementos de salida y comprueba cuál de los tres supera en número de veces el umbral(T):

```
[Y,Pf,Af] = sim(net,vXSim);
r = [];
for i = 1:3
    cont = 0;
    x = Y(i,:);
    for j = 1:1024
        if (x(j) >= T)
            cont = cont + 1;
        end
    end
    r = [r,cont];
end
[val, idx] = max(r);
```

6.3. MODELOS SIMBÓLICOS PARA DESCRIPCIÓN

A diferencia de los métodos o modelos conexionistas en los que se utilizaba una red neuronal bio-inspirada, en los métodos simbólicos prescindiremos de tal abstracción para determinar la existencia de una marca de cantero en un sillar. En esta situación emplearemos métodos basados en la algorítmica matemática y en la estadística descriptiva [8].

6.3.1. CORRELACIÓN ESTADÍSTICA

El primero de los métodos utilizados para la descripción de marcas de cantero se basa en la aplicación de ecuaciones estadísticas para estimar la fiabilidad de la comparación de dos marcas. De esta forma el procedimiento a seguir consiste en obtener un conjunto discreto de patrones de marcas de cantero a comparar. Dicho conjunto de patrones se convertirán en imágenes en blanco y negro, es decir, que los elementos de los vectores estarán comprendidos entre 0 y 1 ($x_i, y_i \in \{0,1\}$). Afortunadamente, como los métodos estadísticos no requieren de la misma complejidad que los neuronales, las dimensiones de las imágenes se puede ampliar a tamaños mayores. En el caso de la presente investigación se ha optado por dimensiones de 100 x 100, con lo que el vector lineal contendrá un total de 10000 elementos.

Como en el caso de las redes de Hopfield, la aplicación consiste en la determinación de 3 marcas básicas:

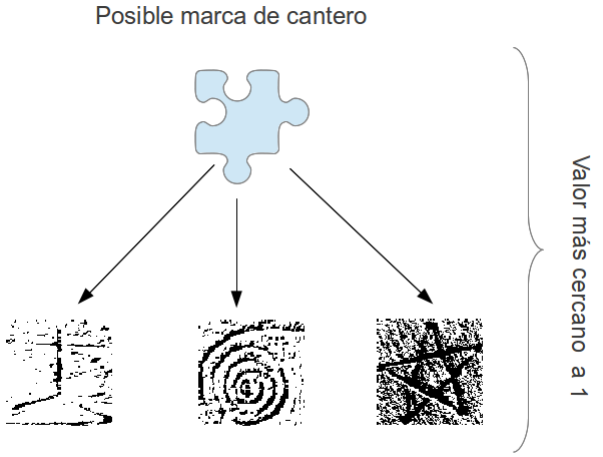


Figura 6.43: Proceso básico de descripción

[8] Cuando nos referimos a métodos simbólicos hacemos referencia a la utilización de notación científica para la descripción de un determinado método de descripción

Por esta razón, una vez capturada la imagen objeto de análisis de un determinado muro, se procede también a binarizarla con el fin de proceder a realizar los siguientes cálculos estadísticos:

Si se supone que entre dos variables existe una relación lineal entonces:

$$S_x = \sqrt{\frac{\sum_{i=1}^{N} x_i^2}{N} - \bar{x}^2} \tag{6.18}$$

$$S_y = \sqrt{\frac{\sum_{i=1}^{N} y_i^2}{N} - \bar{y}^2} \tag{6.19}$$

Donde S_x y S_y corresponde a la desviación típica de x e y respectivamente.

$$S_{xy} = \frac{\sum_{i=1}^{N} x_i \cdot y_i}{N} - \bar{x}\bar{y} \tag{6.20}$$

Donde S_{xy} corresponde a la *covarianza* entre x e y.

Por lo tanto, si suponemos que la variable x_i es la marca patrón y la variable y_i la marca objeto de estudio, el *coeficiente de correlación lineal* se calcula como:

$$r = \frac{S_{xy}}{S_x S_y} \tag{6.21}$$

El coeficiente de correlación lineal toma valores de [-0.3,0.3] para correlaciones débiles; [-0.7, -0.3] y [0.3,0.7] para correlaciones medias; [-1,-0.7] y [0.7,1] para correlaciones fuertes.

Dicho de una forma más algorítmica y aplicada en Matlab, obtenemos el siguiente código:

```
% Funcion para obtener el coeficiente de correlacion lineal

function r = coeficiente(x, y, N)
    Sxy = 0;
    Sx  = 0;
    Sy  = 0;

    for i=1:N
        Sxy = Sxy + x(i)*y(i);
        Sx = Sx + x(i)^2;
        Sy = Sy + y(i)^2;
    end


    Sxy = Sxy/N - mean(x)*mean(y);   % Covarianza
    Sx = sqrt(Sx/N - mean(x)^2); % Desviacion tipica X
    Sy = sqrt(Sy/N - mean(y)^2); % Desviacion tipica Y

    r = Sxy / (Sx*Sy);
end
```

Puesto que las marcas de cantero están muy deterioradas la mayoría de ellas, los coeficientes obtenidos en la aplicación de los juegos de prueba son de un orden <0.1, por lo que para obtener mayor precisión en el reconocimiento se aplica un algoritmo de obtención de la correlación en 0º, 90º, 180º y 270º, de forma que el valor obtenido se acumula en cada iteración. El máximo resultante más próximo a 1 ó a -1 será el candidato elegido.

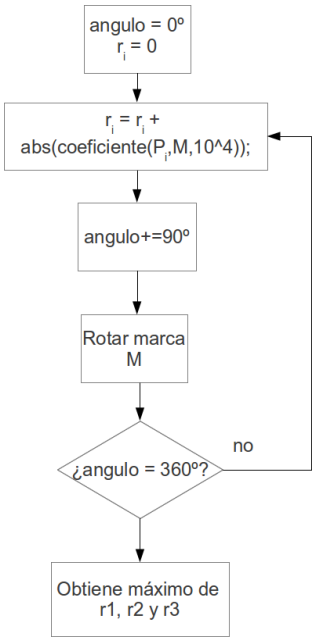


Figura 6.44: Diagrama de flujo del algoritmo de correlación

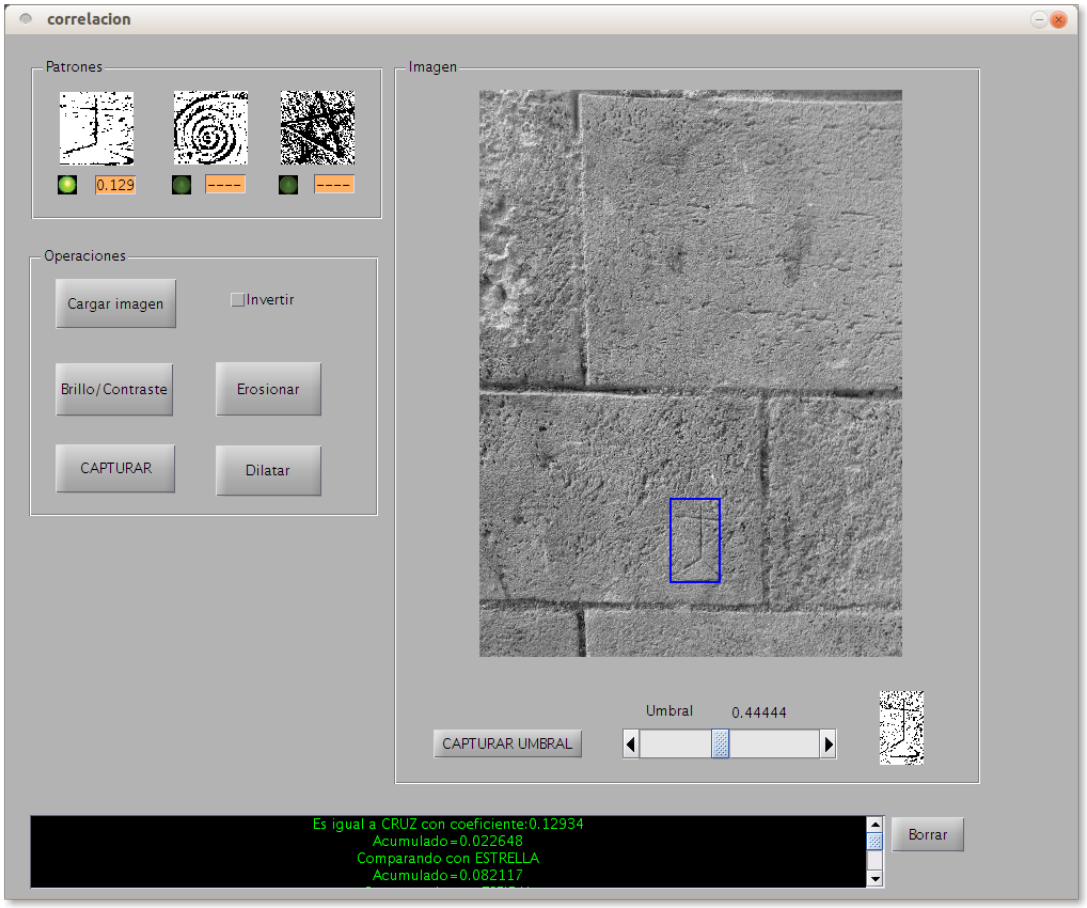


Figura 6.45: Ventana de la aplicación para el algoritmo de correlación

En la figura 6.45 se puede apreciar cómo el algoritmo detecta una marca de cantero en la parte exterior de la Catedral de Murcia obteniendo el máximo del coeficiente de correlación lineal de la imagen binarizada en la parte inferior derecha con el patrón de la parte superior izquierda.

6.3.2. CÓDIGOS DE CADENA

Los códigos de cadena se utilizan para representar una frontera como un conjunto de segmentos con longitud y dirección especificadas. Las representaciones de códigos de cadena suelen tener una conectividad de 4 u 8. Así para una conectividad de 4 se barajan la siguientes posibilidades (N,S,E,O) mientras que para un conectividad 8 se barajan las siguientes posibilidades (N,NE,E,SE,S,SO,O,NO).

Para obtener el código de cadena se divide la imagen en un cuadrícula regular. Después se selecciona una celda aleatoriamente. Si la celda contiene una cantidad en un intervalo determinado a partir de la media se selecciona dicha casilla. En el caso de la investigación se procede a realizar una división de la imagen en cuadros de tamaño 25 x 25, una vez la imagen objeto de estudio se ha capturado y reducido a un tamaño de 100 x 100. Por lo tanto cada celda del grid de análisis contiene un total de 4 x 4 píxeles. De esta forma si dicha celda se representa mediante una matriz:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Se realiza la media de la celda de referencia como:

$$\bar{\alpha} = \frac{a_{11}+a12+...+a44}{16}$$

A partir de esta celda de referencia se comienza a procesar el código de cadena mediante el siguiente algoritmo de búsqueda en profundidad:

```
1
2  % Algoritmo recursivo para la extraccion de codigos de cadena
3  function [] = codigo(x,y,ancho,alto,media,T,vX,cadena, numero)
4
5      % Matriz de visitados
6      global visitado;
7      % Almacena codigo de cadena
8      global codigoCadena;
9
10     % Obtiene media de la zona 4x4
11     m = mean2(vX(y:y+3,x:x+3));
12
13     if ((m >= media - T) && (m <= media + T) && (visitado(fix(y/4) + 1,fix(x/4) + 1)==0))
14
15         visitado(fix(y/4)+1,fix(x/4)+1) = 1; % Marca casilla como visitada
16
17         rectangle('Position',[x,y,4,4],'EdgeColor','b','LineWidth',1);drawnow;
18         codigoCadena = [codigoCadena, numero];  % Acumula codigos de cadena
19
20         if (y + 4 < alto) % Direccion Sur
21             codigo(x, y + 4, ancho, alto,media,T,vX,'S-',1)
22         end
23
24         if (x + 4 < ancho) % Direccion Este
25             codigo(x + 4, y, ancho, alto,media,T,vX,'E-',2)
```

```
26          end
27
28          if (x - 4 > 0)  % Direccion Oeste
29              codigo(x - 4, y, ancho, alto,media,T,vX,'O-',3);
30          end
31
32          if (y - 4 > 0)  % Direccion Norte
33              codigo(x, y - 4, ancho, alto,media,T,vX,'N-',4);
34          end
35
36          if (x + 4 < ancho && y + 4 < alto)  % Direccion Sureste
37              codigo(x + 4, y + 4, ancho, alto,media,T,vX,'SE-',5);
38          end
39
40          if (x + 4 < ancho && y - 4 > 0)   % Direccion Noreste
41               codigo(x + 4, y - 4, ancho, alto,media,T,vX,'NE-',6);
42          end
43
44          if (x - 4 > 0 && y + 4 < alto)  % Direccion suroeste
45               codigo(x - 4, y + 4, ancho, alto,media,T,vX,'SO-',7);
46          end
47
48          if (x - 4 > 0 && y - 4 > 0)      % Direccion noroeste
49              codigo(x- 4, y - 4, ancho, alto,media,T,vX,'NO-',8)
50          end
51
52      end
53  end
```

Es decir, si dado un umbral de tolerancia (T) y la media de la celda de búsqueda ($\bar{\beta}$) tal que:

$$\bar{\alpha} - T <= \bar{\beta} <= \bar{\alpha} + T$$

se aceptará esa celda como válida y se procederá a estudiar el restos de las celdas adyacentes.

Típicamente los valores más frecuentes para T para una imagen en tonos de grises oscilan entre 28 y 40.

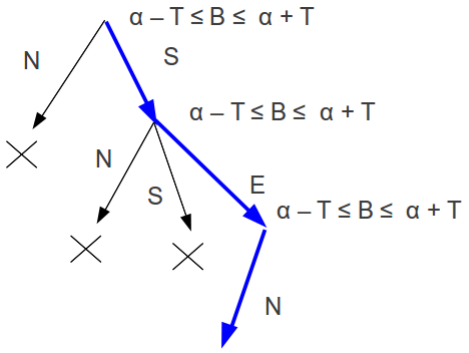


Figura 6.46: Recorrido típico del algoritmo de códigos de cadena

6.3.3. COMPARACIÓN Y UBICACIÓN

Una vez extraídas correctamente los códigos de cadenas durante la fase de descripción, el paso siguiente consiste en comparar los códigos de cadena con los almacenados en la base de datos.

Para la implementación de la base de datos se puede utilizar a un simple directorio indexado de ficheros .txt de códigos de cadena o recurrir a un sistema gestor de bases de datos como MySQL o Microsoft Access. En el caso de la investigación se ha optado por fichero de datos de Matlab denominados ficheros .mat.

Si suponemos que lo implementamos como un directorio indexado de códigos de cadena el algoritmo sería el siguiente:

Para el algoritmo de comparación de los propios códigos de cadena recurriremos al apartado 16.2.2.2 del libro de referencia [1] el cual explica que para comparar dos cadena C_1 y C_2 codificadas de forma $a_1a_2a_3...a_n$ y $b_1b_2b_3...b_n$. Tomamos N_c como el número de correspondencias entre dos cadenas, apareciendo una correspondencia en la posición j si $a_j = b_j$. El número de símbolos sin correspondencia viene dado por:

$$N_{sc} = max(|C_1|, |C_2|) - N_c$$

donde $|C|$ es la longitud de la cadena (número de símbolos) de la cadena C. Se puede demostrar que $N_{sc} = 0$ si y solo si C_1 y C_2 son idénticas:

Una medida sencilla de similitud entre las cadenas C_1 y C_2 se define como el cociente:

$$R = \frac{N_c}{N_{sc}} = \frac{N_c}{max(|C_1|,|C_2|) - N_c}$$

donde $R = \infty$ para una correspondencia perfecta y cero cuando no hay ninguna correspondencia de los símbolos C_1 y C_2. Obviamente los valores cercanos a cero nos indicarán una correspondencia pobre.

Finalmente, una vez encontrada una buena correspondencia se procederá a consultar en la base de datos para encontrar las posibles ubicaciones de la marca en los diferentes talleres de la península ibérica.

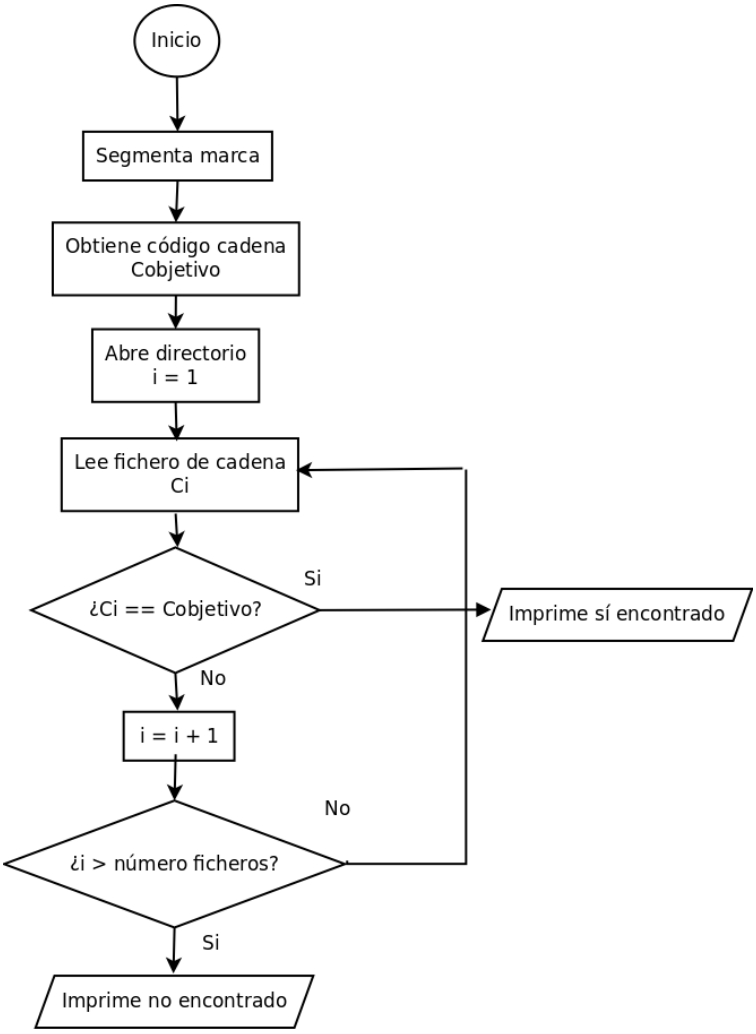


Figura 6.47: Comparación de códigos de cadena

No obstante, se han diseñado dos fórmulas de estimación del mejor y el peor caso de comparación de forma que el investigador pueda tener una mejor referencia de las dos cadenas a estudiar. Por lo tanto, si consideramos que:

$$máximo = max(|C_1|, |C_2|)$$

y

$$mínimo = min(|C_1|, |C_2|)$$

una estimación pesimista sería:

$$Pesimista = \frac{N_c}{máximo} \times 100$$

y una estimación optimista sería:

$$Optimista = \frac{N_c}{mínimo} \times 100$$

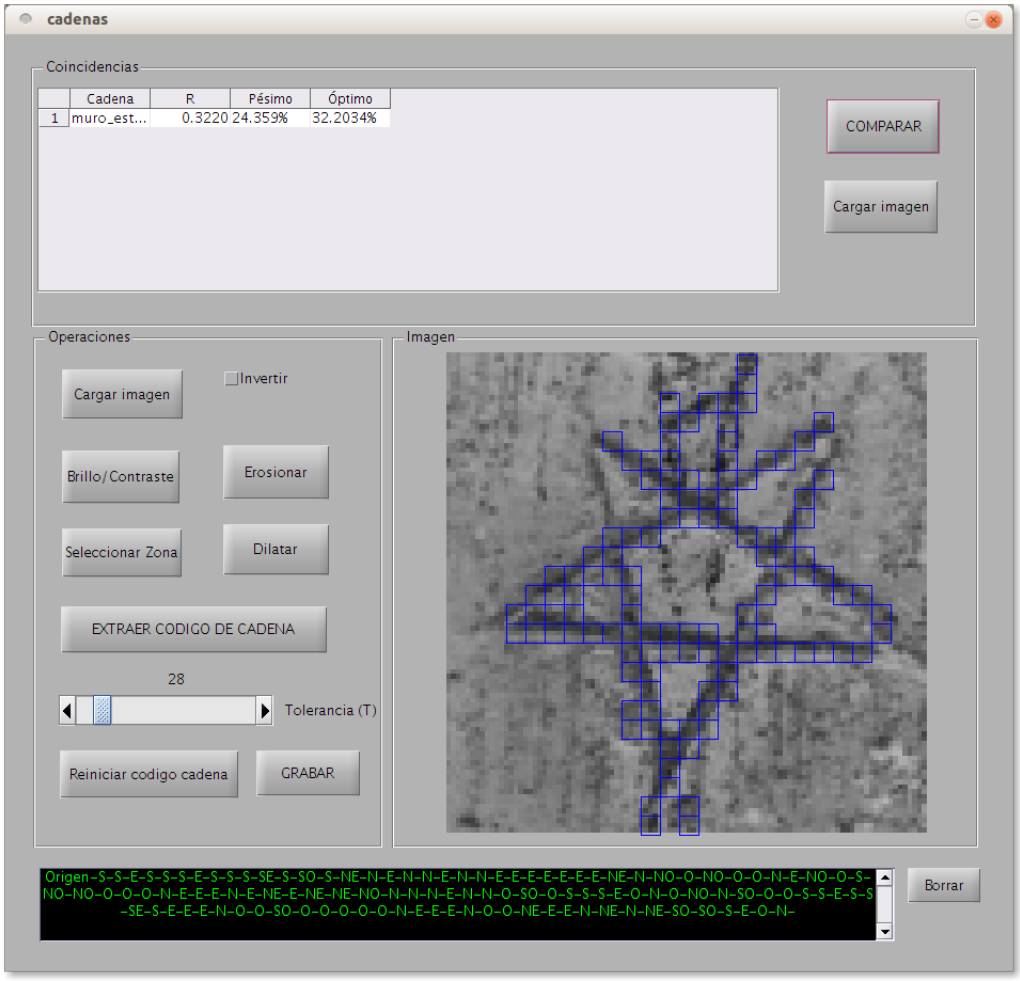


Figura 6.48: Demostración de la captura de un código de cadena

7. COMPARACIÓN DE MODELOS Y EFICACIA

Una vez vistos los diferentes métodos de detección y clasificación de marcas de cantero, procederemos a realizar medidas cuantitativas para evaluar la eficacia de cada modelo en relación con los otros. Para llevar a cabo esta tarea someteremos a cada demostración de cada modelo a un juego de varias pruebas con 4 ensayos cada una. De esta forma concluiremos el porcentaje de eficacia como:

$$P_{hopfield} = P_{multiclase} = P_r = \frac{ensayo_1 + ensayo_2 + ensayo_3 + ensayo_4}{4} \times 100 \tag{7.1}$$

Para detección en Hopfield, en redes multiclase y parte de correlación estadística (P_r), mientras que para el resto utilizaremos la siguiente fórmula:

$$P_{biclase} = \frac{ensayo_1 + ensayo_2 + ensayo_3 + ensayo_4}{\text{coeficiente óptimo} \times 4} \times 100 \tag{7.2}$$

Donde *coeficiente óptimo* será 0.9 para redes neuronales.

También se aplicará la media aritmética a este juego de pruebas.

7.1. EFICACIA DE HOPFIELD

Para realizar un juego de pruebas a las redes de Hopfield se ha procedido a definir la detección de la marca como aceptada (1) o rechazo (0). A partir de la serie numérica binaria se procede a optener los valores de P_{hop} y la media.

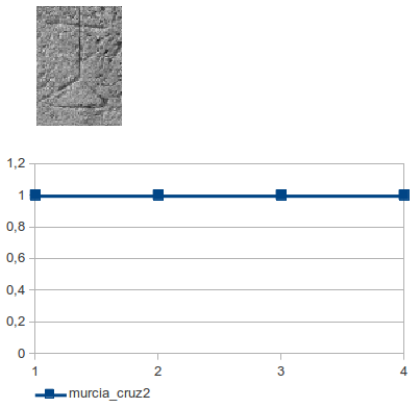


Figura 7.1: P_{hop} = 100 %;Media = 1

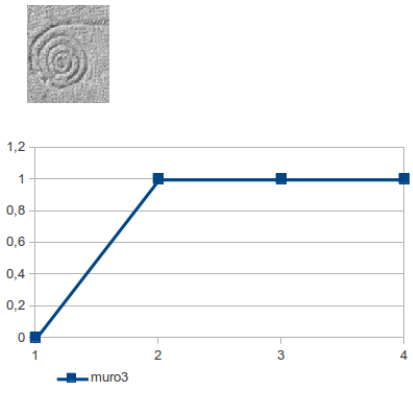


Figura 7.2: P_{hop} = 75 %;Media=0.75

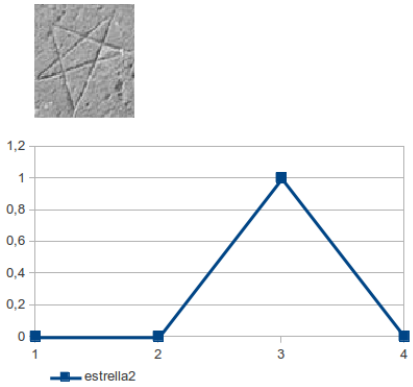


Figura 7.3: P_{hop} = 25 %;Media = 0.25

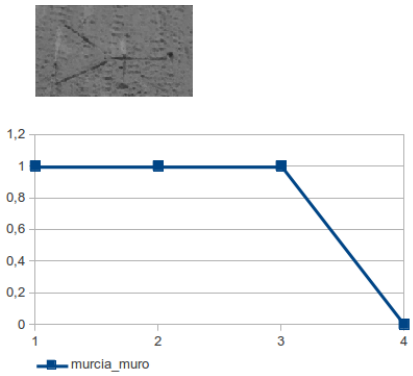


Figura 7.4: P_{hop} = 75 %;Media=0.75

Como se puede apreciar, el modelo de Hopfield tiene un comportamiento aceptable sobre todo cuando se erosionan la imágenes previo a la prueba. En las imágenes se ha utilizado un nivel de tolerancia del 80 %. En la gráfica el valor 1 indica que ha habido éxito, mientras que un valor 0 indica que el algoritmo no ha podido determinar la marca en un determinado ensayo.

En conclusión se puede afirmar que para el algoritmo de Hopfield se tiene una eficacia media total del: 68.75 %.

7.2. Eficacia Red Neuronal biclase

A diferencia de las medidas realizadas a la red de Hopfield, donde se utilizaban valores binarios para indicar éxito o fracaso, en las redes neuronales del tipo backpropagation utilizaremos valores decimales para obtener las tasas P_{bic} y media.

En las redes neuronales biclase se ha procedido a entrenar una red con los siguientes parámetros:

- **Función entrenamiento:** Trainscg (Scaled conjugate gradient backpropagation)
- **Tasa de aprendizaje:** 0.4
- **Epochs:** 400
- **Show:** 10
- **Goal:** 10^{-5}
- **Rendimiento post-entrenamiento:** 0.000682 en epoch 5

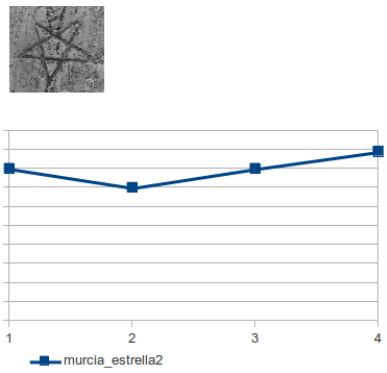


Figura 7.5: P_{bic} = 88 %;Media = 0.79

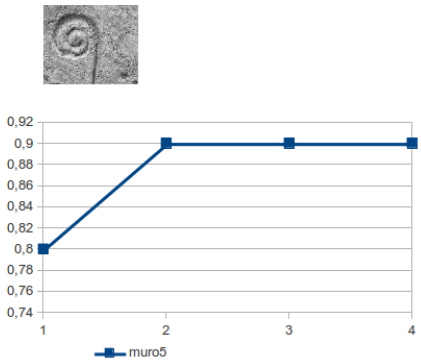


Figura 7.6: P_{bic} = 97 %;Media=0.87

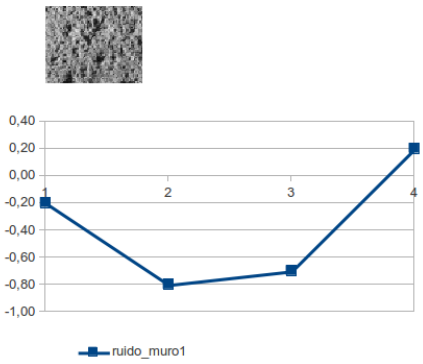


Figura 7.7: P_{bic} = −41 %;Media = -0.37

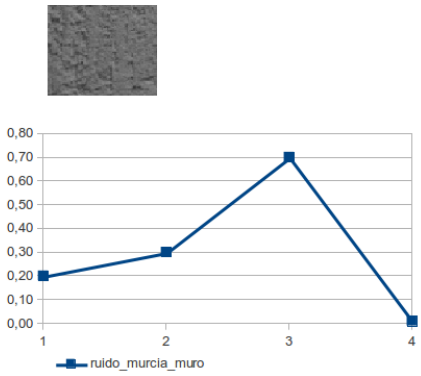


Figura 7.8: P_{bic} = 33 %;Media=0.30

La media total de eficacia para la detección de marcas de cantero en redes neuronales es del 92.5 %, lo que se puede afirmar que tiene buena precisión para la detección de las mismas. Sin embargo, en lo que respecta a la detección de ruido de sillar no obtenemos los mismos resultados, siendo el total del rendimiento un -4 %, de lo que se puede extraer que la eficacia es muy mala. Esto puede ser debido a una falta de entrenamiento exhaustivo en la red neuronal o por la propia naturaleza de iniciación aleatoria de pesos de la red.

Diagrama Box-Whisker

Utilizaremos un diagrama Box-Whisker (“Caja-Bigote”) para representar y comparar de manera intuitiva el conjunto de datos relacionados con las marcas y el ruido. A partir de la información recabada anteriormente se obtienen los siguientes diagramas Box-Whisker mediante Matlab:

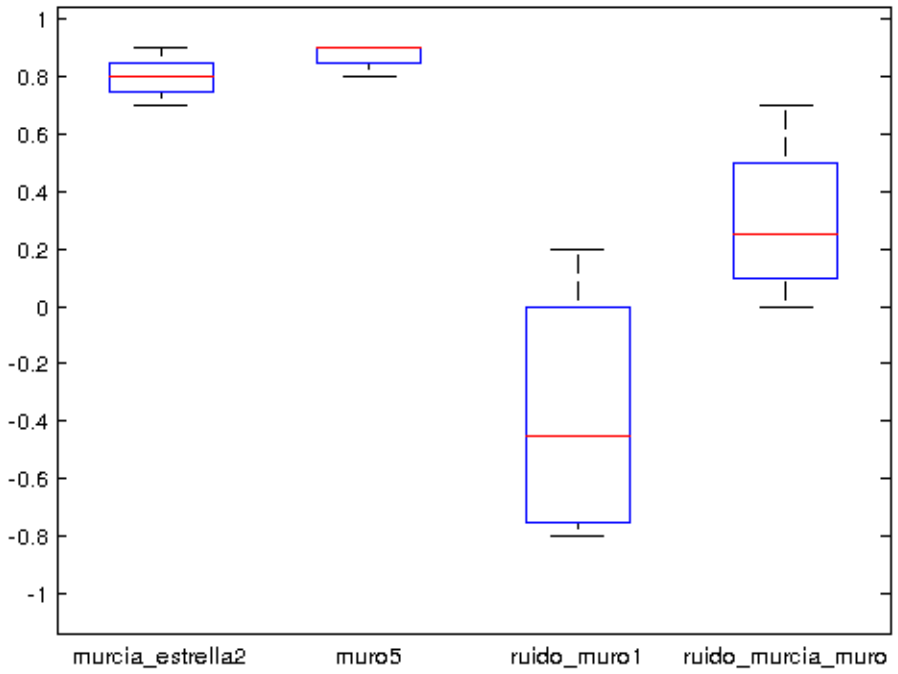


Figura 7.9: Diagrama Box-Whisker de los datos obtenidos

En la figura se representa las diferentes cajas para cada muestra analizada. De esta forma se aprecian en azul los cuartiles 2º y 3º (la caja) que representan el 50% de los datos. Obviamente los datos están ordenados previamente de menor a mayor. La línea en rojo representa la mediana que se define como el valor central de los datos observados. Más precisamente, y puesto que es un conjunto de datos pares, se calcula como:

$$mediana = \frac{dato[\frac{N}{2}]+dato[\frac{N}{2}+1]}{2}$$

Los datos que están fuera de las cajas azules se representan en los "bigotes"que se extienden desde el menor hasta el mayor de los datos observados y por tanto considerados normales.

El déficit de reconocimiento de muestras de ruido en la red neuronal se puede apreciar en los bigotes superiores de la penúltima caja (ruido _muro1) y la última caja completa(ruido _murcia _muro).

7.3. EFICACIA RED NEURONAL MULTICLASE

El caso de las redes neuronales multiclase es el mismo que el modelo de Hopfield, excepto que la determinación de una marca de cantero depende de un valor de tolerancia (T) que se especifica a la aplicación para regular un nivel mínimo de aceptación. Para realizar las medidas también se utilizarán valores binarios.

En el test de pruebas realizado a la red neuronal se han aplicado los siguientes parámetros:

- **Función entrenamiento:** Trainscg (Scaled conjugate gradient backpropagation)
- **Tasa de aprendizaje:** 0.4
- **Epochs:** 4000
- **Show:** 10
- **Goal:** 10^{-5}
- **Rendimiento post-entrenamiento:** 0.000682 en epoch 5

Los resultados han sido los siguientes:

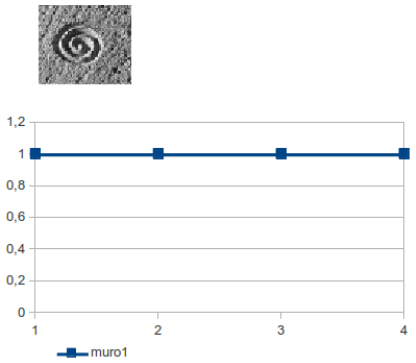


Figura 7.10: P_{mult} = 100 %;Media = 1

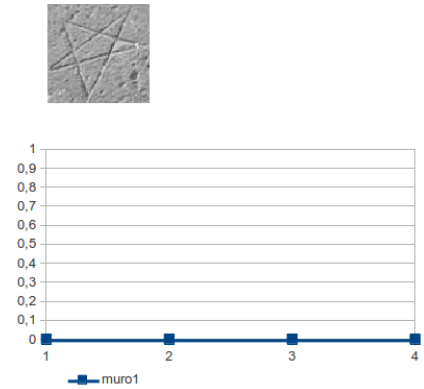


Figura 7.11: P_{mult} = 0 %;Media=0.0

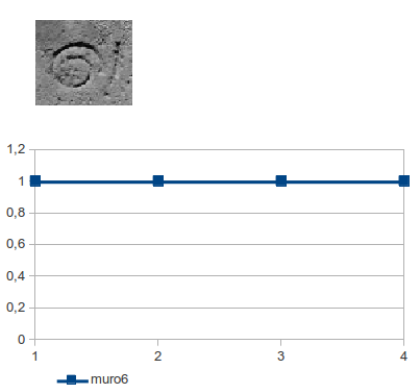


Figura 7.12: P_{mult} = 100 %;Media = 1

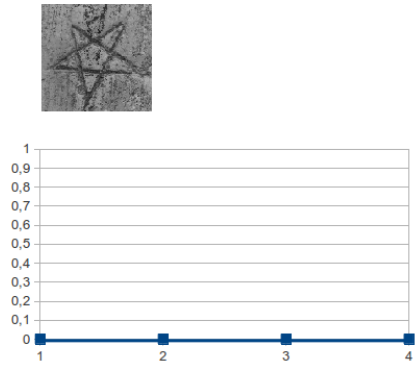


Figura 7.13: P_{mult} = 0 %;Media=0.0

Como se puede apreciar, las redes neuronales multiclase son efectivas para unas determinadas marcas de cantero, mientras que para otras se muestran con nulos resultados. En general, la tasa media total de efectividad de las redes neuronales multiclase son aproximadamente del 50 %.

7.4. Eficacia de la Correlación Estadística

Con la correlación estadística se calculará la efectividad de acierto del algoritmo mediante valores binarios en la serie de (P_r).

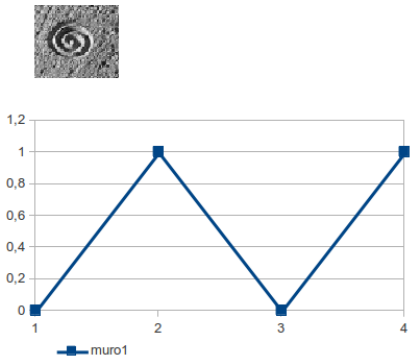


Figura 7.14: P_r = 50 %;$Media_r$ = 0,2

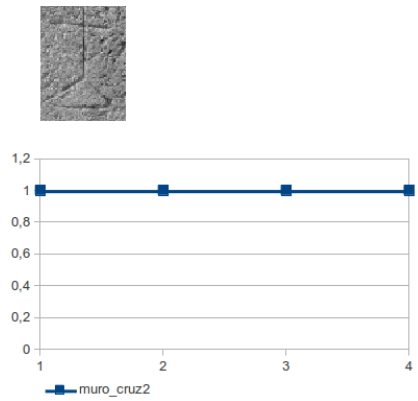


Figura 7.15: P_r = 100 %;$Media_r$ = 0,6

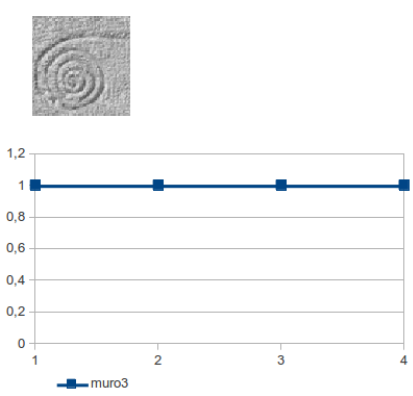


Figura 7.16: $P_r = 100\%$;$Media_r = 0{,}21$

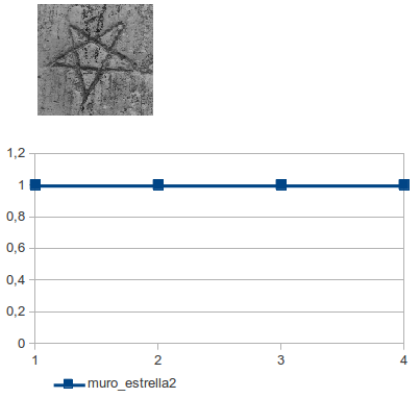


Figura 7.17: $P_r = 100\%$;$Media_r = 0{,}27$

Este modelo presenta mejor eficacia que los presentados anteriormente, si bien necesita que la captura de la marca objetivo sea los más ajustada posible al cuadro de selección con el fin de que la correlación sea más fina. De no ser así el modelo puede presentar falsos negativos, aunque debidos muy seguramente a una falta de coincidencia de la captura con el patrón. Finalmente, la efectividad total del modelo oscila en torno al 85 %, mientras que para la media de los coeficientes de correlación $Media_r$ es 0.32. Como comentábamos anteriormente, la afinación del valor de $Media_r$ depende sobre todo de la capacidad del investigador para ajustar la correlación en la captura de la imagen.

7.5. EFICACIA DE LOS CÓDIGOS DE CADENA

El estudio de los códigos de cadena requiere más complejidad debido sobre todo a la dificultad para extraer el código de forma completa. Además, otro problema subyacente a la comparación de códigos de cadena es la dificultad para encontrar una coincidencia debido, entre otros factores, a que el usuario o investigador puede comenzar la extracción desde diferentes posiciones de la marca.

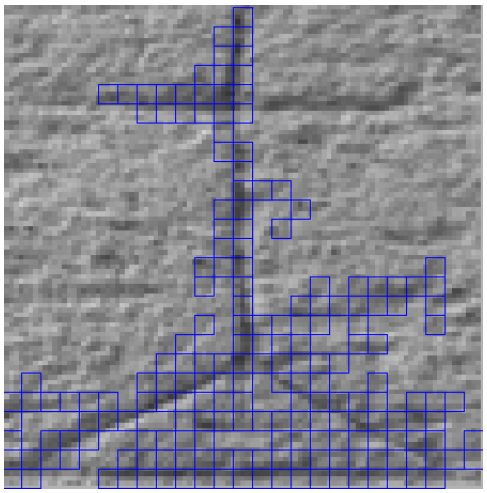


Figura 7.18: Id: muro_cruz3

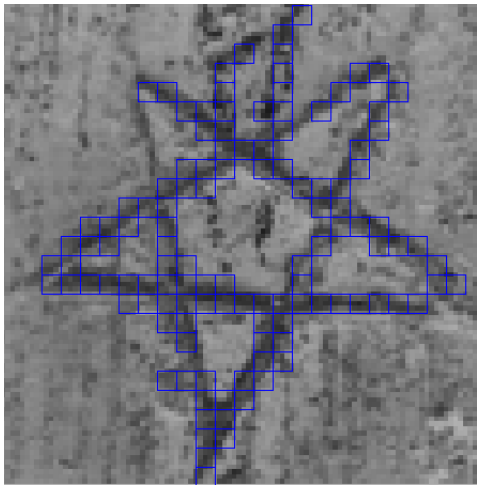


Figura 7.19: Id: muro_estrella2

De esta forma al extraer el código de cadena para la figura 7.18 obtenemos el siguiente resultado al comparar:

ID	Descripción	R	Pésimo	Óptimo
muro_cruz3.mat	Cruz catedral de Murcia	∞	100 %	100 %
muro_estrella2.mat	Pentagrama	0.1059	9.5 %	11.5 %

Cuadro 7.1: Resultados para 7.18

ID	Descripción	R	Pésimo	Óptimo
muro_cruz3.mat	Cruz catedral de Murcia	0.2129	17.55 %	27.73 %
muro_estrella2.mat	Pentagrama	∞	100 %	100 %

Cuadro 7.2: Resultados para 7.19

Como se puede apreciar, existe una correspondencia perfecta ∞ cuando los puntos de inicio de las extracciones tienen igual posición.

7.6. TABLA DE RESULTADOS

La siguiente tabla recoge de forma resumida y a modo de conclusión los totales de las medidas tomadas en cada modelo:

Modelo	$\mathbf{P_{hop}}$	**Media**
Red de Hopfield	68.75 %	0.68
Modelo	$\mathbf{P_{bic}}$	**Media**
Red neuronal biclase (marca)	92.5 %	0.83
Red neuronal biclase (ruido)	-4 %	-0.035
Modelo	$\mathbf{P_{mult}}$	**Media**
Red neuronal multiclase	50 %	0.5
Modelo	$\mathbf{P_r}$	**Media**
Correlación estadística	87.5 %	0.32

Cuadro 7.3: Resultados para 7.19

A la vista de los resultados expuestos y de la investigación aplicada se podría concluir que el mejor método para la comparación de marcas de cantero son los modelos simbólicos, especialmente el método de correlación estadística, siempre y cuando se ajusten los contornos de selección de la captura. En segundo lugar, y siempre que los modelos neuronales estén correctamente iniciados y con un buen rendimiento son también apropiados para la detección biclase de marcas de cantero. Por último, el método de Hopfield también resulta eficaz, pero siempre que se trabaje con imágenes pequeñas y de forma binaria.

En general el mejor método es el de correlación estadística que junto con un algoritmo de eliminación de ruido de sillar puede generar buenos resultados. Finalmente, en todos estos métodos es necesario que la extracción de la marca de cantero de una imagen sea lo más ajustada posible a un modelo de patrón base, en lo que respecta a la delimitación de la captura.

8. TECNOLOGÍA Y HERRAMIENTAS YA EXISTENTES

En mi búsqueda por los medios a los que tengo acceso, no he podido encontrar ningún programa conocido a través del escrutinio en inglés y español en Google. Tan solo en la búsqueda de "gliptografía informática"ha aparecido una referencia, en la página 40, al libro de la editorial Akal sobre "los Canteros Medievales".

En relación al espigado en bibliografía sobre el tema, en el libro base citado sobre marcas de cantero [11] hace una reseña en la página 52 sobre un estudio que se hizo en el año 1984 por J.T. Ramírez y Barberó y J.M. Altozano Foradada en **"La informática, una valiosa herramienta técnica para el estudio de la Gliptografía"**, proporcionan un sistema para desarrollar por medio del ordenador las posibilidades de un banco de datos con todo lo referente a un signo lapidario: distribución, familias de signos, correlaciones, posiciones, frecuencia, etc.

En relación a la Web, recientemente se ha creado la página *Signos lapidarios* [16] donde personas de diferentes localizaciones en la península ibérica recopilan de forma desinteresada marcas de cantero de toda nuestra geografía.

Y también otro estudio, pero menos computacional, atribuido a Ansoáin Simeón Hidalgo Valencia, licenciado en Geografía e Historia por la UNED el cual realizó un estudio de 1500 marcas de cantero distribuidas por toda la geografía de la provincia de Navarra, examinando 125 edificios "piedra a piedra". [9]

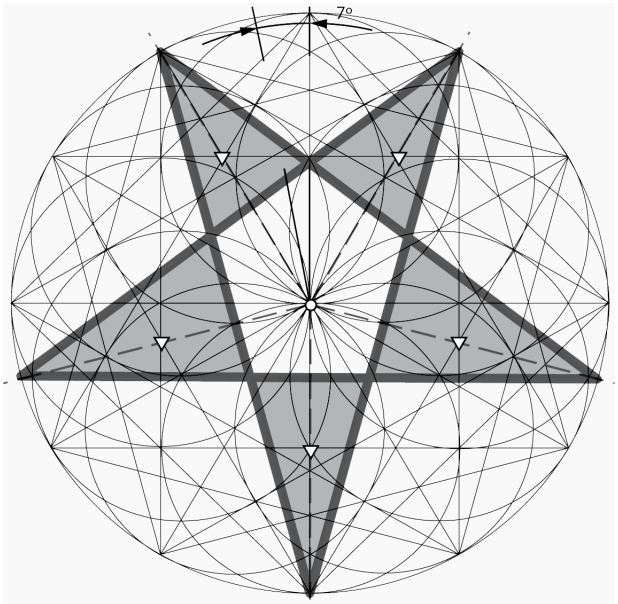
Por lo tanto, a partir de esta criba preliminar de herramientas software se atisba la existencia de aplicaciones de bases de datos para albergar la información de atributos asociados a la marca, sin ninguna alusión a otro software para el análisis digital (segmentación o descripción) de marcas de cantero.

9. CONCLUSIONES

Hasta aquí la investigación realizada durante los seis primeros meses de 2014 de acuerdo a lo estipulado en el reglamento del TFM. Se ha intentado ajustarse a la realidad en la mayor medida de lo posible; aunque es obvio que la segmentación y descripción de las marcas de cantero es un tema complejo que requiere de una revisión de las técnicas aquí propuestas. Espero que este trabajo siente las bases para una futura mejora de los algoritmos de detección de Marcas de Cantero y que sirva como precedente y advertencia de lo que se debe y no se debe utilizar para este tipo de análisis de imágenes. El patrimonio histórico-artístico de nuestro país es un preciado legado heredado de nuestros antepasados y sobre todo, una enseñanza ética y estética de las ideas y creencias de una forma de vida que a pesar de sus circunstancias tenía gran valor para aquellas gentes. Su no conservación y aprecio por parte de las generaciones futuras podría acarrear graves consecuencias debido al desprecio a valores profundamente y elevadamente humanos y espirituales.

[9] Canteros románicos por los caminos de Navarra.ISBN: 978-84-613-3555-8

Finalmente agradecer al profesor *D. Carlos Cerrada Somolinos* la atención que ha tenido durante todo el curso para resolverme dudas y su amabilidad para asistir a Madrid a la demostración del prototipo.



Santo Domingo de Silos_025
(Álvaro Rendón Gómez marzo 2013)

10. BIBLIOGRAFÍA

REFERENCIAS

[1] Gonzalo Pajares Martinsanz, Jesús M. de la Cruz García. *"Visión por Computador. Imágenes digitales y aplicaciones. 2ª Ed."*, Ra-ma, 2007.

[2] Gonzalo Pajares Martinsanz, Jesús M. de la Cruz García. *"Ejercicios Resueltos de Visión por Computador."*, Ra-ma, 2007.

[3] Gonzalo Pajares Martinsanz, María Guijarro Mata-García *"Transparencias del curso de Percepción Visual"*, UCM, 2013.

[4] Roger S. Pressman *"Ingeniería del Software: Un enfoque práctico 5ª Ed."*, Mc Graw Hill, 2002.

[5] Michael Kuhn. *"Manual for the implementation of neural networks in MATLAB"*,Grin, 2005.

[6] Omid B. Sakhi. *"Face detection in Matlab. A practical guide to face detection using Gabor features and Neural Networks"*, 2013.

[7] Alberto García Serrano. *"Inteligencia Artificial: Fundamentos, práctica y aplicaciones"*, Editorial RC, 2012.

[8] M.R. Spiegel. L. Abellanas. *"Fórmulas y Tablas de Matemática Aplicada"*, McGraw Hill, 1991

[9] Lima Díaz, Felipe. *"Manual Avanzado de Java."*, Anaya, 1998

[10] René Guénon. *"Símbolos Fundamentales de la Ciencia Sagrada."* Paidós Orientalia. 1995.

[11] Juan Luis Puente López. *"Firmado en la Piedra"*, 5ªEd. Edilesa.

[12] Xavier Musquera. *"Ocultismo medieval"*, Ed. Nowtilus, 2009.

[13] https://es.wikipedia.org/wiki/Investigacion

[14] Aedo Cuevas. I., Díaz Pérez. P. et al, *"Sistemas Multimedia Análisis, Diseño y Evaluación"*, Servicio de Publicaciones UNED, 2004.

[15] Castro Gil M.A., Colmenar Santos A. , Losada de Dios P., Peire Arroba J., *"Diseño y Desarrollo Multimedia: Sistemas, Imagen, Sonido y Vídeo"*, Ra-ma, 2002.

[16] http://www.signoslapidarios.org/

[17] Murray R. Spiegel, Larry J. Stephens. *"Estadística"*, Editorial McGrawHill, 2009.

[18] Rafael Romero Villafranca, Luisa Rosa Zúnica Ramajo *"Estadística"*, Editorial Universidad Politécnica de Valencia, 1993.

11. LISTADO DE SIGLAS, ABREVIATURAS Y ACRÓNIMOS

CC: Creative Commons.
UNED: Universidad Nacional de Educación a Distancia.
ECTS: European Credit Transfer and Accumulation System.
PC: Personal computer.
3D: Tres dimensiones.
GIMP: GNU Image Manipulation Program.
LCD: Liquid Crystal Display.
RGB: Red, Green, Blue.
JPEG:Joint Photographic Experts Group.
PNG: Portable Network Graphics.
MAX: Máximo.
MIN: Mínimo.
RNA: Red Neuronal Artificial.
XOR: OR Exclusivo.
OCR: Optic Character Recognition.
CAPTCHA: Completely Automated Public Turing test to tell Computers and Humans. Apart.
MSE: Mean Squared Error.
N: Norte.
S: Sur.
E: Este.
O: Oeste.
NE: Noreste.
SE: Sureste.
NO: Noroeste.
SO: Suroeste.
NC: Número de símbolos con correspondencia.
NSC: Número de símbolos sin correspondencia.
TFM: Trabajo fin de Máster.